PUSH-TOWED OCEAN GOING TUG-BARGE OPERATION
IN THE INTEGRAL AND DROP-AND-SWAP MODES:
AN ECONOMIC COMPARISON USING A COMPUTER MODEL

by

JONATHAN DAVID KASKIN

B.S.E.E., University of Pennsylvania
(1971)

S.M., Harvard University
(1977)

SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE
DEGREES OF

MASTER OF SCIENCE IN SHIPPING AND SHIPBUILDING MANAGEMENT

and

OCEAN ENGINEER

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

(JUNE 1979)

Signature of Author..
Department of Ocean Engineering, May 1979

Certified by..
Thesis Supervisor

Accepted by...
Chairman, Department Committee on Graduate Students

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE **JUN 1979**	2. REPORT TYPE	3. DATES COVERED **00-00-1979 to 00-00-1979**

4. TITLE AND SUBTITLE **Push-Towed Ocean Going Tug-Barge Operation in the Integral and Drop-and-Swap Modes: An Economic Comparison Using a Computer Model**	5a. CONTRACT NUMBER 5b. GRANT NUMBER 5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER 5e. TASK NUMBER 5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Massachusetts Institute of Technology,77 Massachusetts Avenue,Cambridge,MA,02139-4307**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S) 11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

Currently, all large ocean-going, mechanically linked tug-barge units are being operated in the integral or ship like mode where the tug and barge remain together at all times. This method of operation does not take advantage of the inherent flexibility available to tug-barge units--the ability to separate the propulsive unit (tug) from the cargo unit (barge). This separability can be used to operate the tugs and barges in a drop-and-swap mode where the tug drops off a loaded barge at a port and then swaps it for one that has completed its cargo operations. This method substantially increases tug utilization since the tug is not required to await cargo operations. However, this increased tug utilization is achieved at the expense of the extra barge units which remain in port for cargo operations. The question is whether in drop-and-swap operation the increased tug utilization outweighs the capital costs of the additional barges. To help answer this question, a computer model is developed to compare the total capital and operating cost of providing transport capacity for simple port pair trades with tugs and barges operated in the integral and drop-and-swap mode. From the model's output it can be determined which mode is more economical for a given trade. The model uses exhaustive enumeration to select among all feasible alternatives the number of tugs and barges, the barge size and form, and the tug-barge system speed that provides the desired transport capacity for a given port pair trade at the lowest required freight rate. The required freight rate calculation takes in account the tug, barge storage tank, and terminal facility annual operating and annualized capital costs. Most of the capital and operating cost elements are obtained from the literature. However practically no data are available for two of the most important cost factors, barge capital construction cost and tug-barge fuel expense. Therefore, two computer subprograms a barge design model and a tug-barge powering model, are developed to estimate these costs. The barge design model subprogram uses the ABS offshore barge rules to determine the midship scantling configuration of single-skin tank barges that results in the minimum hull steel weight as a function of barge size and form. The hull weight estimate is used to predict barge cost. Tabular and graphical output from the model is presented for barges of various size and form. In addition, regression equations for the model output are provided. The tug-barge powering subprogram uses full-bodied bulbous-bowless, single-screw tank ship propulsion and resistance data as an approximation to tug-barge hydrodynamic performance. These data are used in conjunction with a propeller design program to determine the horsepower required to propel the tug-barge unit of a given size and form at a given speed. From the horsepower requirements the fuel expenses can be predicted.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**376**	

PUSH-TOWED OCEAN GOING TUG-BARGE OPERATION

IN THE INTEGRAL AND DROP-AND-SWAP MODES:

AN ECONOMIC COMPARISON USING A COMPUTER MODEL

by

JONATHAN DAVID KASKIN

Submitted to the Department of Ocean Engineering on May 11, 1979, in partial fulfillment of the requirements for the Degrees of Master of Science in Shipping and Shipbuilding Management and Ocean Engineer

ABSTRACT

Currently, all large ocean-going, mechanically linked tug-barge units are being operated in the integral or ship like mode where the tug and barge remain together at all times. This method of operation does not take advantage of the inherent flexibility available to tug-barge units--the ability to separate the propulsive unit (tug) from the cargo unit (barge). This separability can be used to operate the tugs and barges in a drop-and-swap mode where the tug drops off a loaded barge at a port and then swaps it for one that has completed its cargo operations. This method substantially increases tug utilization since the tug is not required to await cargo operations. However, this increased tug utilization is achieved at the expense of the extra barge units which remain in port for cargo operations. The question is whether in drop-and-swap operation the increased tug utilization outweighs the capital costs of the additional barges. To help answer this question, a computer model is developed to compare the total capital and operating cost of providing transport capacity for simple port pair trades with tugs and barges operated in the integral and drop-and-swap mode. From the model's output it can be determined which mode is more economical for a given trade.

The model uses exhaustive enumeration to select among all feasible alternatives the number of tugs and barges, the barge size and form, and the tug-barge system speed that provides the desired transport capacity for a given port pair trade at the lowest required freight rate. The required freight rate calculation takes in account the tug, barge, storage tank, and terminal facility annual operating and annualized capital costs. Most of the capital and operating cost elements are obtained from the literature. However, practically no data are available for two of the most important cost factors, barge capital construction cost and tug-barge fuel expense. Therefore, two computer subprograms, a barge design model and a tug-barge powering model, are developed to estimate these costs.

The barge design model subprogram uses the ABS offshore barge rules to determine the midship scantling configuration of single-skin tank barges that results in the minimum hull steel weight as a function of barge size and form. The hull weight estimate is used to predict barge cost. Tabular and graphical output from the model is presented for barges of various size and form. In addition, regression equations for the model output are provided.

The tug-barge powering subprogram uses full-bodied, bulbous-bowless, single-screw tank ship propulsion and resistance data as an approximation to tug-barge hydrodynamic performance. These data are used in conjunction with a propeller design program to determine the horsepower required to propel the tug-barge unit of a given size and form at a given speed. From the horsepower requirements the fuel expenses can be predicted. Since both subprograms take in account barge size and form, the model determines the most economical form for barges of a given size and speed with respect to capital construction and fuel costs.

The output of the model is presented for five base cases and six sensitivity cases. From these outputs, it is determined that the following trades can be more economically operated in the drop-and-swap mode than the integral mode: 1) trades with large annual cargo flow requirements compared to the ton-mile capacity of the maximum allowed size barge, 2) trades with slow loading/discharging rates that result in voyages with long port times, and 3) trades with substantial terminal storage costs.

Complete documentation for the economic model and for the barge design and power subprograms is provided. The documentation includes program listings, flowcharts, and dictionaries of program variables.

Name and Title of Thesis Supervisor: E. G. Frankel, Professor of Marine Systems.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

ACKNOWLEDGEMENTS

I wish to take the opportunity here to express my appreciation to those individuals who have provided me guidance and assistance during the preparation of this thesis. First I would like to thank Mr. Gregory F. Doyle and Prof. J. Gomes de Oliviera for their technical assistance with the barge design model; Prof. M. Triantafyllou for his guidance on the barge powering model and for the use of his propeller design model; and Dr. J. W. Devanney, III, for his technical advice on the drop-and-swap economic model. I would also like to thank Prof. H. S. Marcus for his assistance throughout this research. Additionally, I express my grateful appreciation of the efforts of Prof. E. G. Frankel and J. A. Brogan of the Military Sealift Command for sponsoring the research presented here since without their efforts this thesis could never have been accomplished. Finally, I would like to thank Carol Bishop and my wife Min Jeung Wu for their dedicated efforts in putting the tables and figures in finished form.

CHAPTER 1

INTRODUCTION

1.1 General Remarks

All of the push-towed ocean going tug barge (OGTB) systems in current use operate in a ship-like integral mode where the tug and barge always remain together. This method of operation does not make use of the only long term advantage that these vessels have over ships, their ability to separate the tug propulsive unit from the cargo barge unit. It is in this drop-and-swap mode of operation that OGTBs should flourish. Therefore, it is the purpose of this thesis to investigate for which port pair trades that this feature of separability can be economically advantageous. This investigation is done through the use of a parametric optimization computer model that compares the cost of operating a port pair trade with tugs and barges in either mode.

The details of this model and the associated barge design and tug-barge powering subprograms are presented in the next three chapters. The reader who is not interested in the detailed assumptions used in the model may proceed to model

results presented in Chapter 5 after having read the remainder of this introductory chapter.

1.2 Background Material on Ocean Going Tug-Barge Systems

The idea of a towboat or tugboat pushing a barge or integrated group of barges is not a new one. For several decades, the river towboat operators have been taking advantage of the better control and lower drag resistance provided by push-towing over pull-towing a barge on a hawser. In the sixties coastwise tug and barge operators, seeing the obvious advantages of push towing, began constructing their new large barges (some over 10,000 DWT) with stern notches for the tug to push in. At first, the notches in these first generation ocean-going tug-barges (OGTBs) were rather shallow so that push-tow operations were restricted to the calm waters of bays, harbors, and gentle seas. Later, these notches became deeper and sophisticated cable or chain linking and chafing gear were used so that push-tow operation of these second generation OGTBs could be extended to all but severe seas. But, it was not until the early seventies (1) that the technology

(1) In the early fifties George G. Sharp, Inc., did design the Car-Port/G1, a tug-barge with a rigid wedge type linkage which was the prototype of the Breit/Ingram design now in use. This rather small system (less than 5000 DWT) ran into serious difficulty with USCG manning authorities which eventually demanded that the tug be manned the same as a ship. This forced the system into foreign-flag service. This tug and barge as well as a follow on tug and two barges operated successfully in a drop and swap mode of operation for many years. It may have been this excessive manning requirement that

was developed for the high horsepower diesel engines that could drive large barges at moderate speeds, and for the reliable linkages that would allow the tug and barge to remain linked together in all types of weather and seas.

Three types of these third generation OGTBs are currently in use in U. S. domestic and foreign trades. They are as follows:

1. Breit/Ingram. In this system the tug is wedged tightly into the barge notch walls and floor so that the linkage is rigid. See Figure 1.1 for a sketch and Hukill (1974) for specific details of this system.

2. Catug. In this system a tongue-like extension from the barge stern is wedged tightly between the catamaran hulls and connecting platform of the tug so that the linkage is rigid. See Figure 1.2 for a sketch and Seabulk (1976) for specific details of this system.

3. Artubar. In this system two large transverse pins (one port and starboard) extend from the tug's bow and fit into sockets located in wing wall skegs that make up the barge stern notch. See Figure 1.3 for a sketch and Fletcher (1974) for specific details of this system.

---

delayed the use of third generation OGTBs until they became large enough so that ship owners, who were not concerned about the ship size manning scale, rather than tug-barge owners began operating them.

A listing with some of the particulars of U. S. flag OGTBs in current or near future operation with the three above designs is given in Table 1.1.

Although only the third generation OGTBs are capable of all-weather push operation, many tug-barge owners have opted to continue construction of the second generation OGTBs. The reason is that these tugs and barges are usually operated on rather short coastal voyages and the fuel that could saved by all-weather rather than just fair-weather push-tow operation does not appear to outweigh the extra manning (2) and linkage (3) costs and reduced flexibility associated with third generation OGTBs. The reduction in flexibility results from third generation OGTB hull forms being not compatible for use with standard tug and barge units. This is because their hull forms having been specially designed to absorb linkage stresses and provide a combined hull form that is ship shape,

---

(2) Under current USCG regulations, the third generation OGTBs which are "mechanically" linked systems must have manning scales determined in the same mannner as ships. The smallest crew certificated for these systems has been about fourteen. On the other hand, second generation OGTBs which are normally chain or cable linked may be manned with far fewer men if the tugs are under 200 GRT. They run with at most ten on voyages of greater than 600 miles and with about seven on shorter voyages. Additionally, only the officer on watch need be licensed.

(3) Third generation OGTB linkages require special reinforcement of the barge notch and tug bow to absorb the stresses resulting from the linkage. This reinforcement as well as sophisticated coupling and decoupling gear can add approximately a million dollars to the cost of tug and barge. Additionally, the rigid linkage systems require heavier scantlings overall, increasing the hull steel weight and cost even further.

not to mate with other tugs and barges.

However, for tugs and barges that are to be normally used in long distance coastal or transocean voyages, the advantages of third generation OGTBs, lower fuel costs due to reduced resistance and better controllability, seem to outweigh the disadvantages; consequently, more operators are building them for these trades. Since the barges have grown to sizes of up to approximately 50,000 DWT, these third generation OGTBs are being used in trades previously served by ships.

There are basically two reasons why third generation OGTBs have been displacing ships, especially tankers and bulkers, in this deadweight size range--they are cheaper to construct and less costly to man than ships. Although it may not seem reasonable that a tug-barge system consisting of two separate vessels and an expensive linkage should be less expensive to construct than the equivalent sized ship, this has been apparently the case. The reason usually given for this anomaly is that when the barge is built in a specialized barge yard (usually with little plate forming capability for the simple barge forms) and the tug in a specialized tug yard (usually without any expensive drydock or building basin), the higher efficiencies resulting from this specialization will outweigh any additional costs due to the linkage. The reduced manning of third generation OGTBs is the result of extreme automation of the tug's diesel plant and the policy that the

barge remain unmanned at sea so that no crew is needed to perform barge maintenance (shoreside personnel perform this work).

To summarize, although it appears that the third generation OGTB is the natural successor to the second generation OGTB, this is true only with respect to its design, not its use. Third generation OGTBs have been embraced by ship operators as a less costly ship replacement and not by the tug-barge operators as a better tug-barge system. (4) The validity of this conjecture is confirmed by the fact that all of the third generation OGTBs have been operated in a ship like mode--that is, the tug and barge always remain together except for overhauls.

It should be noted that the use of third generation OGTBs in place of ships does not seem viable for the long term. If there are significant advantages in building the propulsive and cargo units in separate specialized yards, the same can be done for a ship, as was demonstrated by the construction of the Great Lakes vessel Stewart A. Cort. (5) As

---

(4) Again, this is primarily due to the manning penalty that results under USCG regulations for the "mechanically" linked third generation OGTBs. In fact, the tug-barge operators have been spending their time and energy trying to develop an improved second generation OGTB that will practically achieve the performance of the third generation OGTB with only cable or chain linkages since these systems do not have any manning penalty.

(5) The cargo component was built in a new modular construction yard which was ideal for the cargo unit specialization. The propulsive bow-stern component, however, was built at a

for the lower manning of OGTBs, there is no technical or regulatory reason why ship crews can not be reduced to the same size as third generation OGTBs if they are as fully automated and follow the same maintenance policy. For example, the gas turbine 35,000 DWT Chevron tankers were certificated by the Coast Guard to be manned with fourteen men, the same as most third generation OGTBs. These ships are manned with about twenty men, primarily due to union rules. But, it is expected that these rules will be relaxed when the unions fully realize that they could be forcing operators to use second generation tug-barge systems which are normally manned with fewer non union personnel. So, in the future, a third generation OGTB should not be any cheaper to construct or to run than an equivalent sized ship that is built and operated under the same conditions.

However, there is one inherent feature that third generation OGTBs have in common with all tug-barge systems that is not being taken advantage of by their current users and that cannot be provided by ships. It is the ability to separate the propulsive unit from the cargo carrying unit. This flexibility is commonly utilized in other transport modes, i.e., truck tractor with trailer and train locomotive with rail cars, to increase system efficiency through increased utiliza-

---

large shipyard, probably to ensure that all construction would remain within company owned facilities. So, in this case only partial specialization of facilities was achieved.

tion of expensive propulsive units and through the storage capability of the detached cargo units. It may be that current third generation OGTBs users, being previously experienced only in ship operations, have just overlooked the potential benefits of the separability of their assets, or it may be that it is not economically profitable to utilize this capability in the trades in which they operate.

It is the purpose of the economic model developed in the following chapters to investigate under just what conditions the separability feature of third generation OGTBs should be utilized. If the model indicates that there are few trades where this feature may be used, then it would be expected that ships would eventually push them all-but-out of the shipping scene. However, if the model indicates that there are many trades which could take advantage this feature, then it might be expected that more ships and possibly other transport modes could be replaced by OGTBs in the future. (6)

1.3 Modes of Operation: Drop and Swap versus Integral

As it is the separability feature of OGTBs that make them more versatile than ships, more mention should be made on how this feature can be profitably used. The major benefit is the same as that enjoyed by tractor trailers over single unit

(6) Except for long distance coastal or transocean trades, it would be expected that third generation OGTBs would not displace first or second generation OGTBs until their relative linkage and manning costs can be reduced significantly.

trucks. That is, the propulsive unit (tractor or tug) can be detached from the cargo unit (trailer or barge) while the cargo unit is used for loading, discharging, or storage. It then can be used for transporting another cargo unit that is available for movement. This method of operation, the drop-and-swap mode, obviously increases the utilization of the costly propulsive unit as compared to the integral mode of operation in which the propulsive unit always remains with the cargo unit. However, as can be seen in the simple port pair system shown in Figure 1.4, the drop-and-swap mode will require at least two more barges than tugs in a balanced trade or at least one more barge than tugs in an unbalanced trade (where the tug remains with the barge in one port). Certainly, the drop-and-swap mode would be of most benefit in those trades in which loading/discharging times make up a significant part of the voyage time. Here, then is the most potential in increasing tug utilization, especially in multi-tug fleets which can often be scheduled so that a tug arrives with a barge for discharge just at the time when a barge in port has completed its cargo operations and is available for transport. In this case tug utilization can approach 100%. Also, in trades with long port times, the barges remaining in port in the drop-and-swap mode of operation are used essentially as floating warehouses, replacing shoreside assets. Since port time is a function of the barge cargo capacity and both terminals' loading and discharging rates, and since sea

time is a function of port separation distance and OGTB speed, these four parameters are critical in determining whether the drop-and-swap mode should be used over the integral mode. Because the cost relationships that are functions of these parameters (e.g., fuel cost is a function of the tug-barge size and form, OGTB speed, and port separation distance) are rather complicated, it is not intuitively obvious when one mode is superior to another. A systematic analysis, such as that performed by a computer, is required to determine where the tradeoff point is for the modes.

1.4 Model Features

The economic model described in detail in the next chapter is used to determine for which trades OGTBs should be used in the drop-and-swap versus integral mode. It is in these trades that take advantage of OGTB separability that these systems should flourish, not in those in which they are used as pseudo ships.

The trades analyzed in the model are of the simple port pair type shown in Figure 1.4. This case was chosen since it is the simplest and is appropriate for many of the bulk trades (repetitive voyages from the same loading port to the same discharging port). This port pair trade can be defined essentially by three sets of parameters: (1) port separation distance, (2) terminal loading and discharging rates, and (3) annual cargo flows between ports.

Given the specifics of the trade, the model then determines the barge size (and form) and OGTB speed that will yield the minimum required freight rate (rfr) for both the integral and drop-and-swap modes (balanced and unbalanced). The rfr is defined as the freight rate that should be charged for a unit of cargo that will recover all capital and operating costs on a present-valued discounted cash flow basis (taxes and depreciation ignored). Although specific details on how these costs are obtained is given in the next three chapters, some mention about the general assumptions used in obtaining them is given here so that the reader can skip to Chapter 5 if he wishes to immediately read the results of the model without the details of its procedures. Specifically, with respect to capital and operating costs the following was assumed:

1. Barge capital costs were assumed to be a direct function of barge hull weight with the addition of outfit cost determined via a regression equation found in Sharp (1975). The hull weight as a function of barge size and form was obtained via regression equations developed from output of the barge design program presented in Chapter 3. This program is applicable to single-skin tank barges joined by a semi-rigid linkage to the tug. These types of barges were used since they are the simplest to model and since they are most prevalent of the large OGTBs in use. The semi-rigid rather than rigid linkage was used since it results in less stringent scantling requirements under ABS rules, and in less barge cost. This

rather complicated approach was taken since no reliable barge cost estimate could be obtained from the little available data on OGTBs. In addition, the variation in hull weight as a function of barge form parameters provided by the subprogram is needed for weighing the capital versus operating cost aspects of a barge form.

2. Tug capital costs was determined via a regression equation found in Sharp (1975) and adjusted to conform with prices reported in recent trade literature and government publications. Since many tugs have been built recently, this approach seemed reasonable. Some adjustment is made for the cost of the linkage, which is not extraordinary for semi-rigid linkages.

3. Storage capital costs were calculated for oil storage tanks. The costs were based on recent cost figures provided by a major oil company.

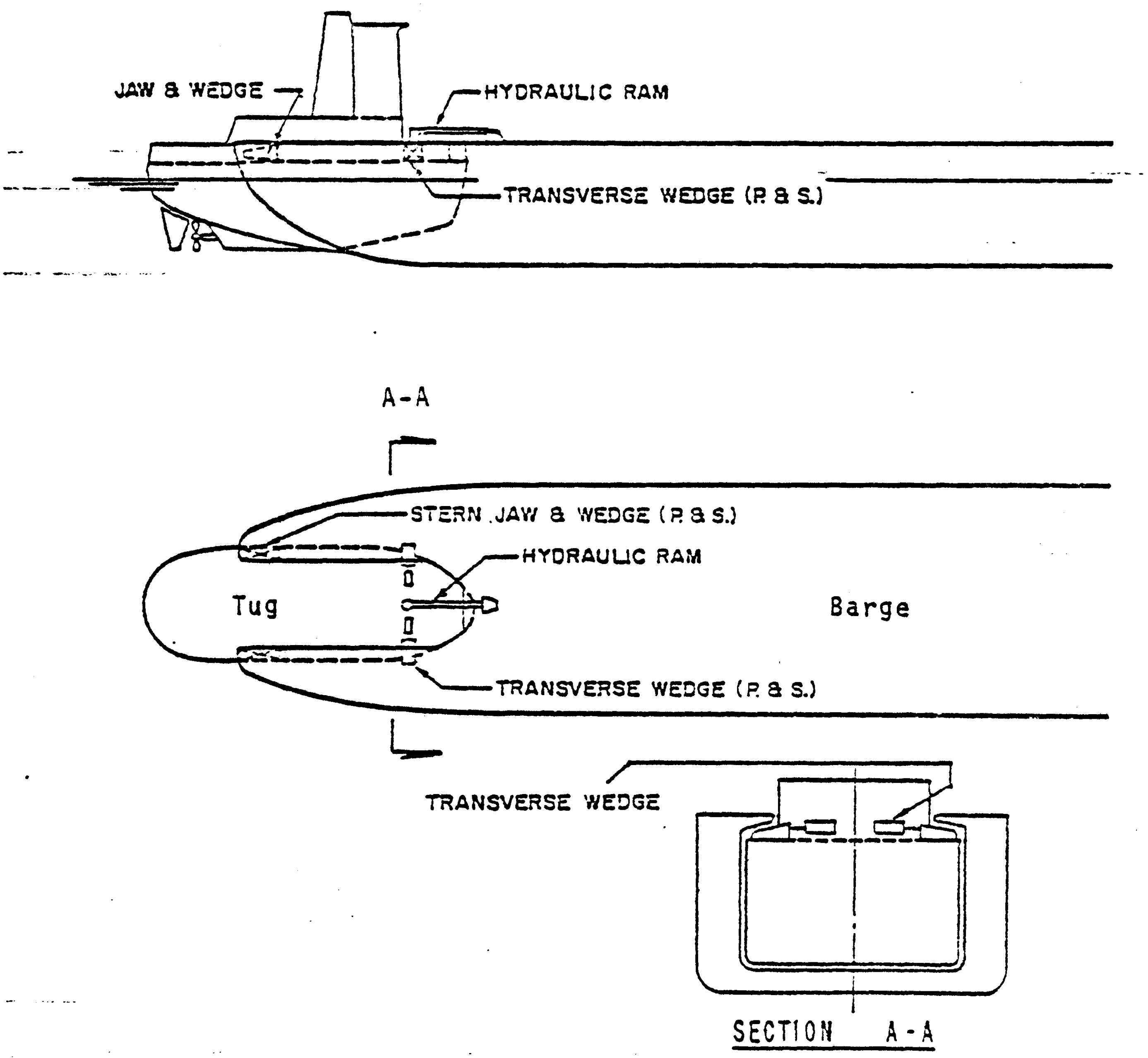
4. Tug fuel costs were determined as a function of tug-barge resistance and voyage duration. Barge resistance, a function of barge form and speed, was determined with the use of full-bodied, bulbous bowless, single-screw tank barge resistance data found in Tsuchida (1969). This was the only series resistance data that could be found to approximate OGTB hull forms. An additional 10% resistance was added to account for linkage interferences to conform with the estimates found in Robinson (1976).

5. Other operating costs were calculated by using the

equations found in Sharp (1975) and then inflating them to yield a current estimate--for 1 January 1979.

Although the economic model has been specifically developed for the tank barge case, it should still be indicative of costs for other bulk trades. Barge outfit cost would probably be the only major change for other trades. Thus, any results obtained from the tank barge model, even for trades with very long port times which are not usual for oil barge trades, should not be in great error.

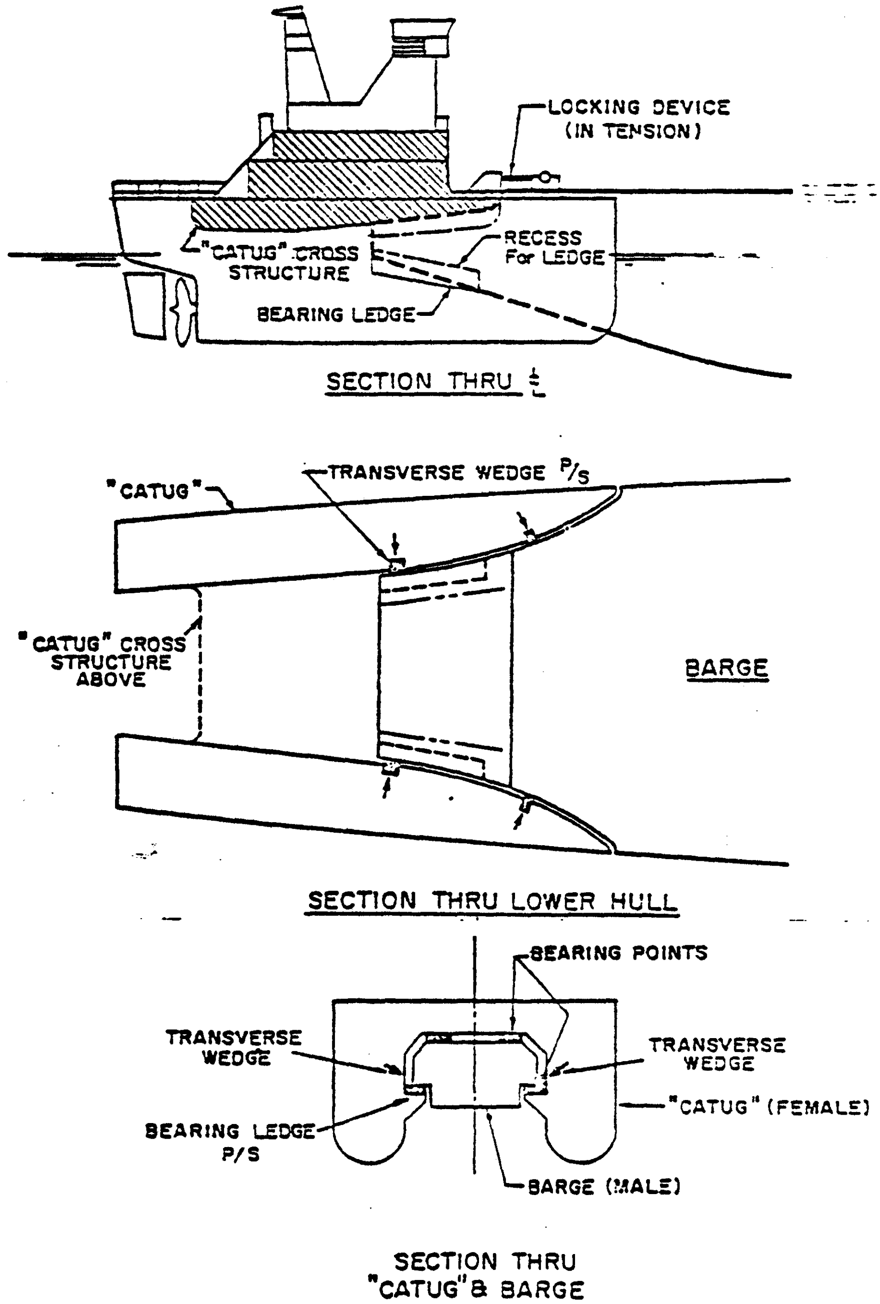
In the following chapters frequent mention will be made to specific computer program variables which will be enclosed in quotation marks. Although their definition will normally be understood by context, detailed definitions of these variables are provided in the appendices.



SOURCE: Waller (1972)

FIGURE 1.1

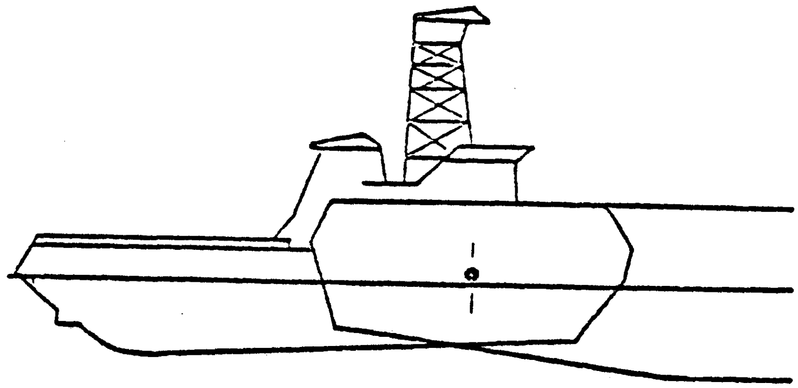
BREIT/INGRAM OGTB LINKAGE DESIGN



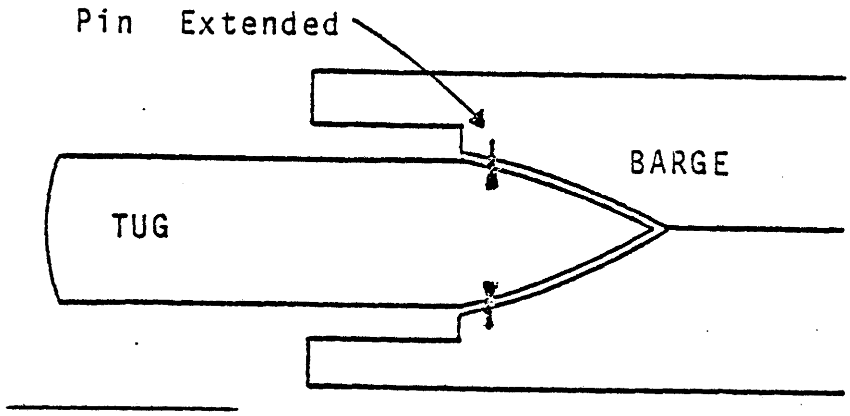
SOURCE: Waller (1972)

FIGURE 1.2

CATUG OGTB LINKAGE DESIGN



Deep Notch-Pinned "Artubar"



From the tug, port and starboard pins are extended toward the barge. The bullet shaped pins aid alighment. In the extended position, the pins fit into lubricated sockets within the barge.

SOURCE: Waller (1972)

FIGURE 1.3

ARTUBAR OGTB LINKAGE DESIGN

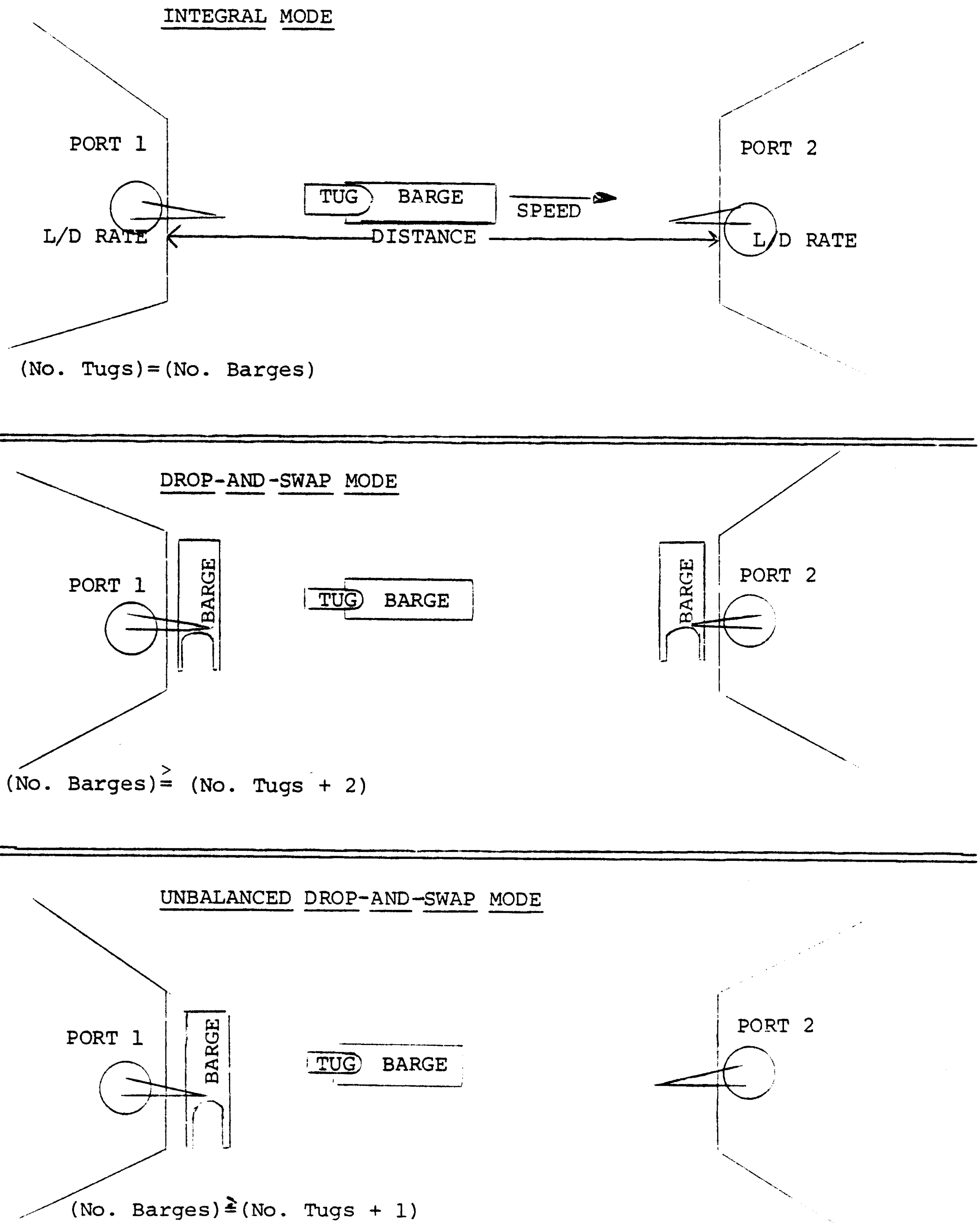


FIGURE 1.4
PORT PAIR TRADES: INTEGRAL AND DROP-AND-SWAP MODES

TABLE 1.1

U.S. FLAG THIRD GENERATION OGTB'S

Linkage Design	Barge Type/ Service	Name: Tug Barge	Dimensions (LxBxD): Tug Barge	Tug-Barge: Draft Length	Tug-Barge: HP/Design Speed DWT	When Built: Tug Barge	Where Built: Tug Barge	Owner/ Operator
Breit/Ingram	Taconite Dry Bulk/ Great Lakes	Presque Isle Presque Isle	140.33'x54.0'x31.25' 974.5'x104.58'x46.50	29' 1000'	14,840/16mph 52,000	12/73 "	Halter Marine Erie Marine	Crocker National Bank/ U.S. Steel
Breit/Ingram	Clean Products Tank/Ocean	Martha R. Ingram IOS 3301	145.84'x46.0'x30.25' 584.5'x87.0'x46.33'	37'5" 620'	11,128/14.1 Kts 36,500	7/71 3/71	Southern Shipbuilding Alabama Drydock	Ingram Corp.
Breit/Ingram	Clean Products Tank/Ocean	Carole G. Ingram IOS 3302	145.84'x46.0'x30.25' 584.50'x87.0'x46.33'	37'5" 620'	11,128/14.0 Kts 37,500	3/72 "	Southern Shipbuilding Levingston Shipbuilding	Ingram Corp
Breit/Ingram	Rice-Phosphate Dry-Bulk/Ocean	Valerie F Valerie F	150.67'x54.0'x34.0' 620.0'x85.0'x45.0'	30'8" 656'	16,000/15.5 Kts 25,000	12/76 "	Southern Shipbuilding Maryland Shipbuilding	
Breit/Ingram	Fertilizer Dry-Bulk/Ocean	Jamie A. Baxter CF-1	125.0'x45.0'x27.75' 500.0'x75.25'x46.5'	32' 600'	7200 /12.5 Kts 22,500	6/76 12/77	Peterson Builders Avonsdale Shipyard	C.F. Industries
CATUG	Oil Tank/ Ocean	Seabulk Challenger SSC-3901	116.08'x90.44'x38.42' 591.0'x95.0'x46.0'	37' 629'	14,000/15.5 Kts 35,000	1/75 "	Galveston Shipbuilding (Kelso Marine)	Hvide Shipping/ Shell Oil
CATUG	Chemical Tank/Ocean	Seabulk Magnachem SSC-3902	116.08'x90.44'x38.42' 582.17'x95.0'x52.0'	40'1" 615'	14,000/15.5 Kts 40,000	2/77 "	Galveston Shipbuilding (Kelso Marine)	Hvide Shipping/ Diamond Shamrock
CATUG	Superphosphoric Acid/Ocean	Two or Three Systems to be Named	126'5"x90'4"x39' 626'6"x99 x50'	36' 677'10"	18,200 /15.5 Kts 41,250	80+	Avondale Shipyard	Occidental Oil
CATUG	Oil Tank/ Ocean	Two Systems to be Named	127'7"x90'4"x39' 645'x95.0'x61.6"	40'6" 699'4"	18,200/15.5 Kts 47,075	80-81	Halter Marine Bethlehem Steel	Amerada Hess
ARTUBAR (2-units)	Ro-Ro/ Ocean	JJ Oberdorf Other Names Unknown	140'x40'x 568'x85'x41'6"	19' 605'6"	7,500 6450(554 vehicles)	79-80	Marinette Marine Seatrain Shipbuilding	Coordinated Caribbean Transport

CHAPTER 2

DROP-AND-SWAP COMPUTER MODEL: FORMULATION AND ASSUMPTIONS

2.1 Model Formulation

The drop-and-swap computer model has been developed to make an economic comparison of the operation of push-towed ocean going tug-barge combinations in the drop-and-swap versus integral modes. As shown in the model formulation summary in Figure 2.1, the model makes this comparison by determining for both modes the number of tugs and barges and the barge speed, size (DWT), and form (block coefficient--Cb, length-breadth ratio--L/B, breadth-draft ratio--B/T) that results in the minimum required freight rate for cargo transported on a port pair trade. (1) This is subject to the conditions that the system has sufficient ton-mile capacity to carry the annual cargo flows and sufficient number of terminal facilities at each port to handle the annual port throughput.

The objective function, the required freight rate (rfr), is equal to the system capital costs (tugs, barges, terminal

(1) See Figure 1.4 and Section 1.4 for the definition of a port pair trade.

and storage facilities) divided by a present value factor (1) plus system annual operating costs (fuel costs, crewing costs, storage and terminal costs, etc.) all divided by the annual cargo flows. As shown in Figure 2.2, these capital and operating costs are nonlinear functions of the port pair trade parameters--port separation distance, terminal facility loading/discharging rates, and annual cargo flows--as well as the five continuous and three integral system variables--Barge DWT, Speed, Cb, L/B, B/T; Number of Tugs, Number of Barges--Port 1, and Number of Barges--Port 2.

Since the last three variables must be integral, this makes the model's form a mixed-integer non-linear program. Programs of this type are not amenable for solution by optimization techniques. If they can be solved at all, it is usually by some sophisticated specialized technique that transforms the model to one that is easier to solve but with many more variables. Rather than taking this approach, I decided to use the brute force method of exhaustive enumeration because it is the simplest to program and because it provides good results at reasonable cost when the ranges of the variables to be considered are chosen judiciously. With this approach, I calculated for each port pair trade and for both modes of operation the required freight rate for all possible

(1) The present value factor apportions the capital costs on an annual basis. It is a function of the capital's pre-tax discount rate or rate of return and the economic life of the system.

combinations of the discretized values (1) of the five continuous system variables. Given a specific combination of these variables, the capacity and continuity constraints determine the values of the three integer variables.

Since I used discretized values for the continuous variables, the minimum required freight rate found will most likely not be the true minimum. However, since the objective function was found to very flat near the optimum, the rfr found will be close to the true minimum even if rather large increments for the variables are used. And, of course, a more accurate estimate of the true minimum rfr can be obtained by using smaller increments, although the cost will probably not warrant the additional accuracy achieved.

Prior to proceeding to a discussion of the computer model's logic and assumptions in the next section, mention should be made of the parametric ranges of the system variables for which the model will produce valid results. These ranges, which are governed by the valid ranges of the formulae used in the model, are provided in summary form in Table 2.1.

(1) To limit the number of combinations to a finite and reasonable number, it was necessary to discretize these continuous variables by dividing their parametric ranges into equally spaced increments. The number of increments can vary from two to fifteen depending on the sensitivity of objective function to changes in the variable.

2.2 Summary of Program Logic

A summary of the logic of the drop-and-swap computer program is shown in flow chart form in Figure 2.3. A brief discussion of the overall logic in this section will be followed by a detailed discussion of each step in the next section.

The program begins by asking the user to specify the port pair trades to be considered as well as the parametric ranges and increments of the five continuous system variables. It then asks the user to specify whether he wishes to change any of the semi-fixed parameters that are used in the cost calculations. Following this, it asks the user to specify what form he wants the output. The program then begins the cost calculations for a specific port pair trade iteration defined by port separation distance and terminal loading/discharging rates. Given these port pair trade parameters, the model determines the number of terminal facilities with the specified loading/discharging rate (and the number of barges for the drop-and-swap mode) that are required at each port.

Then the program selects the next iterative values for the four barge size and form system variables--Barge DWT, Cb, L/B, and B/T. Given these values, it calculates the barge length, hull steel weight, cost, and the tug and barge principal dimensions. The program then checks to see if these system variables are feasible in that they fit within the inter-

polation table ranges for the tug-barge residual resistance coefficients. If not, the program skips to the next iteration of the four barge size and form variables. If so, the program selects the next iterative value of tug-barge speed and calculates the horsepower required to be delivered to the propeller of the tug, the tug-barge resistance, and finally the horsepower of the engine required to be installed onboard the tug. Given these values, the cost of the tug is calculated and then the number of tugs required to provide sufficient transport capacity for both drop-and-swap and integral modes. Then the operating costs and the total capital costs are determined for both modes. From these the rfr can be determined. If the rfr is less than that calulated during previous iterations of the five continuous system variables, it is saved; otherwise, it is ignored. After all iterations of the variables are examined, the minimum rfr found is stored for that port pair trade. After all of the port pair trades are examined the program can present the results in various graphical forms.

2.3 Drop-and-Swap Program Detailed Logic and Assumptions

The detailed logic and assumptions used in the drop-and-swap program are shown in flowchart form in Appendix A. It will be useful to refer to these flowcharts in the discussions to follow.

2.3.1 Input of Port Pair Trade and System Variables (Refer to Figure 2.4)

After typing the execution command "drop_and_swap", the program reads from tape into memory the values of the loadline factor, residual resistance coefficient, self-propulsion factor, and propeller design coefficient arrays. The program then asks the user to "Input via list format the following parameters:". (To input via list format, the user types in the value of the requested parameter followed by a comma.) The first request is a question on the desired form of the terminal facility loading/discharging rates: "Do you wish to specify individual L-D rates?". If the user answers negatively ("no", "n", or "0"), then the program requests values for "minrate, maxrate, delrate". This is a request for a range of loading/discharging rates in tons per day per terminal facility to be investigated, from "minrate" to "maxrate" in "delrate" increments. It is assumed that the loading and unloading rates at both terminals will be the same ("rload1"="runload1"="rload2"="runload2"). Also, if "delrate" is set to zero, then a "delrate" equal to 1000 is assumed. If, on the other hand, the user answers affirmatively ("yes", "y", or "1") to the loading/discharging rate question, then the program requests values for "rload1, runload1, rload2, runload2". This is a request for a specific set of terminal facility loading and discharging rates for each port. In

either case, any set of loading/discharging rates may be specified.

Now the program continues with a request for the values of "mindist, maxdist, deldist". This is a request for a range of port separation distances in nautical miles to be considered from "mindist" to "maxdist" by "deldist" increments. If "deldist" of zero is inputted, then a "deldist" of 1000 nautical miles is assumed. Any set of port separation distances may be considered.

Next the program requests values for "minspeed, maxspeed, delspeed". This is a request for a range of tug-barge speeds in knots to be considered from "minspeed" to "maxspeed" by "delspeed" increments. If "delspeed" of zero is inputted, then a "delspeed" of 1.0 knot is assumed. Any set of speeds can be considered. However, a minimum speed of six knots and a maximum of twelve or thirteen knots will usually cover the optimum speed range and will not exceed the boundary restrictions for Froude number and IHP shown in Table 2.1.

Next the program requests values of "mindwt, maxdwt, deldwt". This is a request for a range of barge cargo deadweights in long tons (LT) to be considered from "mindwt" to "maxdwt" by "deldwt" increments. If "deldwt" of zero is inputted, then a "deldwt" of 5000 LT is assumed. Any set of deadweights from 5000 to 80,000 LT can be used.

Next the program requests values for "aflowave1, aflowave2". This is a request for the annual average cargo

flows in long tons from Port 1 to Port 2 and from Port 2 to Port 1, respectively. Any pair of values can be specified, except that "aflowave1" must be greater than or equal to "aflowave2". For example, if a one way trade is desired, then "aflowave2" is set to zero.

Next the program requests values for the three barge form variables, "mincb, maxcb, delcb", "minlb, maxlb, dellb", and "minbt, maxbt, delbt". These three requests are for the ranges of the barge block coefficient (Cb), tug-barge length-breadth ratio (L/B), and barge breadth-draft ratio (B/T) to be considered from "mincb", "minlb", "minbt" to "maxcb", "maxlb", "maxbt" by "delcb", "dellb", "delbt" increments, respectively. If "delcb", "dellb", or "delbt" of zero is inputted, then a value of 0.1, 0.2, or 0.1 is assumed, respectively. The valid ranges for these form parameters are given in Table 2.1. If the user does not input values within these ranges (including the reduced confidence ranges) then the program will ask the user to specify a new set of form parameter ranges. After the program has accepted the form parameter ranges, input of the system variable and port pair trade ranges to be considered has been completed.

2.3.2 <u>Input and Modification of Semi-Fixed Parameters (Refer to Figure 2.4)</u>

Now the program outputs the statement, "Input changes to semifixed data via get data format". This is a request to the user to make any modifications to the semi-fixed parametric

data that is used in the required freight rate calculations. The base case values of these parameters that are read from tape into memory are shown in Figure 2.5. Definitions of these parameters can be found in Appendix A . The user may modify the value of any of these semi-fixed parameters by simply typing the parameter name followed by an equal sign and then followed by the desired parameter value. Each parameter that is modified should be separated by a comma and the final one should be followed by a semicolon.

2.3.3 Selection of Output Format

Next the program asks the user to specify what type of format he desires for the program output. Specifically, it asks, "Do you want printed output?:". If the user answers in the affirmative, the program responds, "Do you want detailed output?:". If the user answers this question in the affirmative, the program will print out a line of output for every single iteration with respect to port pair trade and system variables. An example of this output for the input case shown in Figure 2.4 is shown in Figure 2.6. If the user answers negatively to the detailed output question, the program will print the system data associated with the iteration that resulted in the minimum required freight rate for both modes (drop-and-swap, then integral) for each port pair trade considered. An example of this output for the input case shown in Figure 2.4 is shown in Figure 2.7.

On the other hand, if the user answers negatively to the question about printed output, or after the program has completed printing output, the program will ask, "Do you wish graphic output?". If the user answers negatively, the program will start from the beginning, asking for a new set of inputs. Otherwise, if the user answers affirmatively, the program will ask a series of questions concerning the form of the graphical output. Samples of graphical output are found in Chapter 5.

2.3.4 Iterations With Respect to Port Pair Trades

At this point the program begins the first of its iterative loops. It now selects the next incremental value for the iterative variable "distance" within the range "mindist" to "maxdist". This specifies the port separation distance of the trade under consideration. Next, the program will either use the values of "rload1", "rload2", "runload1", and "runload2" specified at the beginning of the program or will set these variables to the value of the iterative variable "rate" within the range "minrate" to "maxrate". This specifies the terminal facility loading/discharging rates of the port pair trade under consideration.

After this, the program zeroes the arrays ("best1" and "best2") that store the characteristics of the tug-barge systems that have the lowest required freight rate for the given port pair trade. Then it calculates the monthly average cargo flows ("mflowave1" and "mflowave2") which are the annual

flows apportioned on a monthly basis taking in account that the barge is available only "bargeopdays" of the year for service.

2.3.5 Calculation of the Number of Terminal Facilities (and Barges for Drop-and-Swap Mode) Required at Each Port

Now that the port pair trade characteristics have been defined, the program determines the number of terminal facilities with the specified loading and discharging rates that must be located at each port to handle the annual cargo flows. This is also the number of barges that must be handled simultaneously at each port for the drop-and-swap mode of operation. This number is determined by dividing the monthly cargo flows ("mflowave1" and "mflowave2") by the monthly terminal facility throughput capacity (30.5 x L-D rate). The specific formulae used are shown in Table 2.2. It should be mentioned that seven days a week operations were assumed.

2.3.6 Iterations With Respect to Barge Size and Form

Next the program begins the iterative loops with respect to barge size and form. First it selects the next incremental values for the iterative variable "dwt" within the range "mindwt" to "maxdwt". This specifies the barge cargo deadweight to be used in the calculations to follow. Then the program selects the next incremental values for the iterative form variables "cb", "lb", and "bt" within the ranges "mincb" to "maxcb", "minlb" to "maxlb", and "minbt" to "maxbt",

respectively. This specifies the tug-barge Cb, L/B, and B/T to be used in the calculations to follow.

2.3.7 Calculation of Barge Hull Weight and Principal Dimensions

Given the values of the barge DWT and form variables specified in the iterative loops discussed above, the program calculates the barge length (1) and hull weight values via quadratic interpolation with respect to block coefficient from the formulae provided in Table 2.3. These formulae were derived from the output of the single-skin tank barge program discussed in the following chapter.

Next the program determines the barge breadth by dividing the tug-barge length ("litb") by the length-breadth ratio ("lb"). Similarly, the barge draft is determined by dividing the barge breadth by the breadth-draft ratio ("bt"). It should be noted that the tug-barge length is assumed to be equal to the barge length plus seven-tenths the tug length. This assumption closely approximates the lengths of the large articulated push-towed ocean going tug-barge systems now in operation.

Now the program begins a short iterative loop by using a formula found in Sharp (1975) to estimate the barge outfit

---

(1) Barge length used in these formulae refers to the distance at the waterline from two-thirds of the barge notch length forward of the stern to the barge stem. This is in accordance with ABS rules pertaining to articulated tug-barge systems presented in MARAD (1979).

weight as a function of barge length. Sample values from this formula are shown in Table 2.4. The value of outfit weight summed with the cargo deadweight and with the barge hull steel weight is used to estimate the barge displacement. The barge displacement is used, in turn, to obtain an improved estimate of the barge length. This procedure is then repeated iteratively until the barge length as well as barge displacement and outfit weight converge to unchanging values. Given the value of outfit weight, the program calculates an estimate of the barge cost by summing the product of outfit weight times a cost per ton outfit factor (1) with the product of hull steel weight times a cost per ton hull steel factor. (2)

If the barge length is found to exceed 750 feet, which is beyond the feasible range of the model, then the program skips to the next iterative value for "cb". Otherwise, the program calculates the barge freeboard using the rules specified in IMCO (1966). The specific formulae and table values used are shown in Table 2.5 and Figure 2.8. It should be noted that in the freeboard calculations it was assumed that the barge had no sheer and that the barge is unmanned so that a 25% reduction in freeboard is allowed. After the barge

(1) This factor is assumed to be $12,820 per long ton outfit. This is based on the value found in (Sharp, 1975) inflated by 30% to bring it a January 1979 value.

(2) This factor is assumed to be $1100 per long ton hull steel. This is based on 40 man-hours per ton at $15 per man hour (including overhead) plus $500 per ton material cost.

freeboard is calculated, the barge depth is determined. At this point the program checks to see if the tug-barge unit's dimensions exceed any of the length, breadth, or draft limitations ("maxl", "maxb", "maxt1", or "maxt2") that may have been specified in the semi-fixed parametric data for the port pair trades under consideration. If any of these limitations are exceeded, the program skips to the next "bt" iteration. Otherwise, it checks to see if the form parameters "cb" and "lb" are within the table interpolation ranges specified in Table 2.1. If they are not, the program skips to the next "lb" iteration; otherwise, it continues as described in the following section.

2.3.8 Iteration With Respect to Tug-Barge Speed and Calculation of Tug IHP and Cost

At this point the program begins the last of the iterative loops which is with respect to tug-barge speed. It selects the next incremental value for the iterative variable "speed" within the range "minspeed" to "maxspeed". Then the value of "speed" and the tug-barge principal dimensions ("lbarge", "bbarge", "tbarge", and "cb") are fed into the Subprogram "power". This program, described in detail in Chapter 4, returns the value of the horsepower required to be delivered by the propeller ("dhp") to propel the tug-barge system through the water at the specified speed. (1) From

(1) This program also calculates "ehp", the power required to propel the tug-barge system though the water, from which the

this value the shaft horsepower can be determined. It should be noted that a tug shaft efficiency of 98%, an appendage drag of 5%, and a linkage drag of 10% were assumed. (1)

Given the value of the service margin (2) for the tug, the horsepower of the engines required to be installed onboard the tug can be determined. From this the cost of the tug can be estimated using the formulae found in Sharp (1975), inflated 30% to bring them up to a January 1979 level. Sample values from these formulae are presented in Table 2.6.

2.3.9 Calculation of the Number of Tugs Required

Now the program begins the calculations to determine the number of tugs required to provide sufficient movement capacity for the required annual cargo flows. It does this for the drop-and-swap modes, balanced and unbalanced, (3) and then the

---

still water hull resistance can be determined. It also provides values for the self-propusion factors ("wa", "th", and "hr") and open water propeller efficiency "propef".

(1) The assumptions for appendage and linkage drag are based on conversations with articulated tug-barge designers. They seem optimistic when compared to the results presented in Robinson (1966). However, this study used fairly crude prototype linkage forms and so probably overestimated the drag for the modern linkages which are well faired.

(2) In the model's base case the service margin, the additional fraction of "ehp" required to ensure that the service speed is achieved in most seas, is assumed to be 0.20.

(3) The unbalanced drop-and-swap mode ("dsopt" = 1) is the case where it is assumed that the tug will remain with the barge in the port with the shortest time spent for cargo operations. This would be appropriate for one-way trades with short loading and long discharging times. In this case the

integral mode.

To determine the number of tugs required in the drop-and-swap modes, the program first calculates the time required for cargo operations in both ports ("tport1" and "tport2"). For "tport1", this is the barge cargo deadweight divided by the terminal facility loading rate plus the cargo deadweight--weighted by a balance factor if cargo flows are not equal in both directions--divided by the terminal facility discharging rate. A similar formula pertains for "tport2". Then the program calculates the sea voyage time for a round trip. This is twice the distance divided by the system speed plus linking and unlinking times if appropriate plus any other expected port delays. (1) For the drop-and-swap modes, these values are fed into an iterative routine that calculates the total tug voyage time ("ttript") which includes seatime plus any time that tug is required to wait for completion of cargo operations ("twait1" for Port 1 and "twait2" for Port 2). The routine also calculates the minimum number of tugs ("mintug") required for the trades.

The logic of the iterative routine is rather simple.

---

waiting time for the port that the tug remains with the barge is equal to the cargo operations time, i.e., "twait1" = "tport1". The program selects the drop-and-swap mode that results in the lower rfr to be stored and printed.

(1) In the model's base case the port delay and tug-barge linking/unlinking times were estimated to be four hours. The port delay time takes in account the expected time for docking and undocking as well as time awaiting berth for the barge.

After the program assumes initial values for "mintug", "twait1" and "twait2"; it calculates new values for "twait1" and "twait2" based on the number of barges stationed at each port and the currently assumed value for "mintug". This is done by assuming the tugs are on equally spaced time schedules. Given the values of "twait1" and "twait2", the total voyage time "ttript" is determined. Given this value, the minimum number of tugs ("mintug") for the trade can be determined by comparing the required monthly cargo flows with the ton-mile capacity of each tug-barge unit. The program then will iterate and calculate new values for "twait1" and "twait2" until the total voyage time "ttript" converges on an unchanging value. If convergence does not occur, an error message is printed. If it is found that the total port waiting time exceeds the time that would be spent for cargo operations, then the program prints out a message stating that the drop-and-swap mode would not be appropriate. Otherwise, the program continues by calculating annual operating costs, as described in the next section.

To calculate the number of tugs required in the integral mode, a simpler approach is used than for the drop-and-swap mode. In this case, the total voyage time is equal to seatime and port time; and, the number of tugs required is simply the minimum that will provide sufficient flow capacity (number of tug-barges per month times barge cargo deadweight) to handle the monthly cargo flow requirements.

2.3.10 Calculation of Annual Operating Costs

Now that the program has determined (1) the duration of the tug seatime ("seatimet") which includes time for unlinking/linking and port delays, (2) the duration of in port time ("portimet") which includes the time that the tug must await cargo operation completion, (3) tug shaft horsepower ("shp") for achieving the specified speed, and (4) the tug engine size ("ihp"); it is now able to proceed to calculate the various components of the total annual operating cost per tug-barge unit. Discussion of the assumptions used in calculating each of the cost components is provided below.

Annual Diesel Fuel Cost. The annual diesel fuel cost is equal to the number of tug voyages per year ("nrtrips" = "tugopdays"/"ttript") times the amount of diesel fuel in long tons consumed per voyage ("fuelcons") times the cost per long ton diesel fuel. (1) The amount of fuel consumed per voyage ("fuelcons") is equal to the tug at sea time in hours ("seatimet") times the hourly at sea fuel consumption rate in tons per hour ("rseafuel") plus the tug in port time in hours ("portimet") times the hourly in port fuel consumption rate ("rportfuel" = 0.125 ton/hr). The at sea fuel consumption rate ("rseafuel") in long tons per hour, is in turn equal to the product of the diesel engine's specific fuel consumption

(1) In the model's base case, diesel fuel cost is assumed to be $140 per long ton.

rate ("sfc") (1) in pounds per horsepower-hour and the tug's shaft horsepower ("shp"), all divided by 2240.

Annual Lube Oil Costs. The annual lube oil cost is equal to the number of tug voyages per year ("nrtrips") times the amount of lube oil in gallons consumed per voyage ("lubecons") times the cost per gallon for lube oil. (2) The amount of lube oil consumed per voyage ("lubecons") is assumed to be equal to the tug at sea time in hours ("seatimet") times the hourly at sea lube oil consumption rate in gallons per hour ("rlubeoil"). This hourly at sea fuel consumption rate has been assumed to be equal to the tug's shaft horspower divided by 4000., in gallons per hour.

Annual Crew Costs. The annual crew costs are equal to the average annual crew member's wages and benefits ("cwages") plus subsistence expenses ("csubs"), all times the number of crew members onboard the tug ("nrcrew"). (3)

---

(1) In the model's base case, a sfc of 0.36 is assumed. This is a reasonable value for the medium speed diesels currently used in high powered tugs. In the future lower sfc's and fuel costs may be obtained with the use of low speed diesels burning heavy fuels.

(2) In the model's base case, lube oil is assumed to be $1.75 per gallon.

(3) In the model's base case the average crew size has been assumed to be sixteen, which is very close to the minimum manning level of fourteen that the U. S. Coast Guard has previously allowed for "mechanically-linked" push-towed ocean going tug-barges. (The extra two men are used to fill cook/steward positions.) As for the average crew member's wages and subsistence, they were assumed to be $65,000 and $3,500 respectively. These vlues are in reasonable agreement with the Maritime Administration data shown in Table 2.8.

Annual Costs for Maintenance and Repairs, Insurance, and Stores, Supplies, and Equipment. The annual costs for maintenance and repairs ("amandr"), insurance ("ainsur"), and stores, supplies, and equipment ("asupplies") are determined from formulae found in Sharp (1975). These formulae are functions of the tug engine size and total deadweight of the tug-barge unit. Sample values from these formulae, which have been inflated to bring them up to January 1979 levels, are presented in Table 2.7.

Annual Port Charges. The annual port costs are equal to the number of voyages per year times the port charges per voyage. The voyage port charges consist of a fixed charge per port call ("cfixport1" and "cfixport2") plus a variable cost which is a function of the barge size ("cvarport1" x "dwt" and "cvarport2" x "dwt"). (1)

Annual Costs from the Time Value of Cargo. Since the cargo represents a significant capital investment for its owner, the cost of the capital that is tied up while the cargo is being transported should be considered in the total operating costs for the system. The annual cost for the time value of the cargo ("acargo") is equal to the product of the annual cargo flows ("aflowave1" + "aflowave2") times the sea time in years and times the discount rate for capital ("disrate").

(1) In the model's base case, all the port charge factors ("cfixport1", "cfixport2", "cvarport1", and "cvarport2") are assumed to be zero.

Annual Terminal and Storage Operating Costs. The annual terminal operating costs ("atermop") are simply the product of the annual cargo flows ("aflowave1" + "aflowave2") and the average cost per ton cargo for loading/discharging operations ("cvarterm"). Similarly, the annual storage costs ("astorop") are simply the product of the annual cargo flows and the average cost per ton cargo for in port storage. (1)

Calculation of the Total Annual Operations Costs. At this point the total operating costs per tug-barge unit ("aopcost") can be determined. It is simply the sum of fuel, lube oil, crewing and subsistence, maintenance and repair, stores, supplies, and equipment, insurance, cargo value, port charges, and other miscellaneous ("aother") (2) costs. The total operating costs ("totopcost") are then equal to the number of tug-barge units times the operating cost per tug plus the terminal operating costs plus fleet administrative costs ("admin"). (3) For the drop-and-swap modes 3% of the cost of the additional barges required to be stationed at the ports is added to take account for the maintenance and repair, stores, supplies and equipment, and insurance incurred by these addi-

(1) In the model's base case, all terminal and storage costs are assumed to be zero.

(2) In the model's base case it is assumed that the miscellaneious other costs amount to $30,000 per year for each tug.

(3) In the model's base case it is assumed that the administrative costs per fleet amount to $150,000 per year.

tional units. For the integral mode, the annual storage costs are added to take in account that shoreside storage facilities must be used for storage rather than the barges that remain in port for the drop-and-swap modes.

To show that the values resulting from the formulae and other assumptions used in calculating the operating costs are reasonable, a comparison of the model's output is made with some Maritime Admininistration tug-barge data in Table 2.8.

2.3.11 Calculation of System Capital Costs

After the total operating costs are calculated, the program then determines the total system capital cost ("totcapcost") for the drop-and-swap and integral modes of operation. This total capital cost consists of the barge price times the number of barges in the system plus terminal and storage facility capital costs. These costs are adjusted to take account of the increased productivity that result from multi-unit orders by use of the learning curve factor formula found in Sharp (1975). The barge and tug prices are equal to the tug and barge costs ("ctug" and "cbarge") calculated previously, but increased by 10% to take into account shipyard profit. The terminal facility cost is assumed to be some cost factor ("cfixterm") in dollars per ton-day times the sum of the terminal facility loading/discharging rates times the number of facilities per port. The storage facility cost, which pertains to the integral mode only, is assumed to be equal to

the product of some cost factor in dollars per cargo deadweight ton ("cfixstor") times the barge cargo deadweight times the number of terminal facilities ("minbarge1" + "minbarge2"). This formula assumes that a storage facility equal to the barge capacity must be built onshore for each terminal facility. This is only one of many possible ways of estimating shoreside storage requirements and was used just to give an indication of how storage costs might affect the tradeoffs in operating in the drop-and-swap versus integral mode. (1) In a real operating environment, the storage capacity will depend primarily on the ability of the tug-barge systems remaining on rigid schedules.

2.3.12 Calculation of Required Freight Rates

At this point the program has completed all the calculations required to determine the required freight rate for recovering all operating and capital costs for the system. The required freight rate is simply the total system capital cost ("totcapcost") divided by the present value factor ("pvf") plus the total system operating costs ("totopcost"), all divided by the total annual cargo flows. Now all calculations have been completed within the iterative loops. All that remains to be done is the storage within the computer program of the system variable and cost values for the

(1) In the model's base case terminal and storage facility capital costs were assumed to be zero.

iterations resulting in the lowest required freight rate for the port pair under consideration.

2.3.13 Storage of System Parameters for Minimum RFR Iterations

For each iteration of barge size, form, and speed, the program determines the associated required freight rate for the drop-and-swap, unbalanced drop-and-swap, and integral modes of operation. If for a particular iteration the rfr for either regular or unbalanced drop-and-swap mode is found to be lower than that found for previous iterations of the drop-and-swap modes, then the system values for that iteration replace those previously stored in the array "best1". The same pertains to the integral mode which has its optimum system values stored in the array "best2". So, at the end of all the iterations with respect to tug-barge DWT, Cb, L/B, B/T, and speed; the arrays "best1" and "best2" contain the system values for the drop-and-swap mode (the better of the balanced and unbalanced modes) and integral mode, respectively, that result in the minimum rfr. At this point the program will print out (if printed output was requested) the optimum system parameters for both drop-and-swap and integral modes for each trade specified by its port separation distance and loading/discharging rate. It also will store certain of the system parameters such as minimum rfr, optimum barge size and speed, etc., in storage arrays that will be used in the graphical output routines. Following this, the program will

return to the beginning, requesting new inputs.

This concludes the detailed description of the drop-and-swap program. Discussion of the tug-barge powering and barge hull weight subprograms follows in the next two chapters.

Objective Function:

Minimize (Required Freight Rate)

For a port pair trade defined by:

(1) Port separation distance

(2) Terminal facilities loading/discharging rates

(3) Required annual cargo flows

Capacity Constraints:

$$\left\{\begin{array}{l} \text{Ton-mile:} \\ \quad (\text{No. Tugs}) \times (\text{Barge DWT}) \times \left(\dfrac{\text{No. Voyages}}{\text{Tug-Year}}\right) \geq \left(\begin{array}{l}\text{Required Annual} \\ \text{Cargo Flows}\end{array}\right) \\ \\ \text{Terminal Facility:} \\ \quad (\text{No. Facilities per Port}) \times \left(\dfrac{\text{Annual Thruput}}{\text{per Facility}}\right) \geq \left(\begin{array}{l}\text{Required Annual} \\ \text{Cargo flows} \\ \text{Through Port}\end{array}\right) \end{array}\right\}$$

Continuity Constraint:

(No. Barges) = (No.Tugs) +(No. Barges-Port 1) + (No. Barges-Port 2)

Parameter Boundary Constraints:

mindwt	$\leq$	Barge DWT	$\leq$	maxdwt
minspeed	$\leq$	Tug-Barge Speed	$\leq$	maxspeed
mincb	$\leq$	Barge C_B	$\leq$	maxcb
minlb	$\leq$	Tug-Barge L/B	$\leq$	maxlb
minbt	$\leq$	Barge B/T	$\leq$	maxbt

Integrality Conditions

$$\left\{\begin{array}{l} \text{No. Tugs} \\ \text{No. Barges - Port 1} \\ \text{No. Barges - Port 2} \end{array}\right\} \text{integer}$$

FIGURE 2.1

MODEL FORMULATION

$$\text{Required Freight Rate} = \left\{ \frac{\dfrac{\text{Capital Costs}}{\text{PVF}} + \text{Annual Operating Costs}}{\text{Annual Cargo Flows}} \right\}$$

Capital Costs = (No. Barges) x (Barge Cost) + (No. Tugs) x (Tug Cost) + (Terminal Costs) + (Storage costs)

$$\left\{ \begin{array}{ll} \text{Barge Cost} & = f\{\text{Barge DWT, L/B, B/T, } C_B\} \\ \text{Tug Cost} & = f\{\text{Installed Horse power (IP)}\} \\ \quad \text{IHP} & = f\{\text{Barge DWT, L/B, B/T, } C_B\text{, Tug-Barge Speed}\} \\ \text{Terminal Cost} & = f\{\text{Terminal Loading/Discharging Rates}\} \\ \text{Storage Cost} & = f\{\text{Barge DWT}\} \end{array} \right.$$

Annual Operating Costs = Fuel Costs + Terminal Costs
+Storage Costs + M&R Costs
+Crewing and Subsistence Costs
+Insurance Costs + Administrative Costs

Fuel Costs = Seatime x (Sea Fuel Consumption Rates)
+Port time x (Port Fuel Consumption Rate)

$$\left\{ \begin{array}{l} \text{Seatime} = f\left(\dfrac{\text{Port Separation Distance}}{\text{Tug-Barge Speed}}\right) \\ \text{Port Time} = f\left(\dfrac{\text{Barge DWT}}{\text{Terminal Facility L/D Rate}}\right) \\ \text{Sea Fuel Consumption Rate} = f\left\{ \begin{array}{l} \text{Barge DWT, L/B, B/T} \\ \text{Block Coefficient;} \\ \text{Tug-Barge Speed} \end{array} \right\} \\ \text{Port Fuel Consumption Rate} = f(\text{Hotel Load}) = \text{constant} \end{array} \right.$$

Terminal Costs = f(Annual Cargo Flows)

Storage Costs = f(Annual Cargo Flows)

$$\left. \begin{array}{l} \text{Maintenance and Repair Costs} \\ \text{Supplies and Equipment Costs} \\ \text{Insurance Costs} \end{array} \right\} = f(\text{Barge DWT, Tug IHP})$$

$$\left. \begin{array}{l} \text{Crew Wages and Benefits} \\ \text{Crew Subsistence} \end{array} \right\} = f(\text{crew size}) = \text{constant}$$

Administrative Costs = constant

FIGURE 2.2

DEFINITION OF REQUIRED FREIGHT RATE

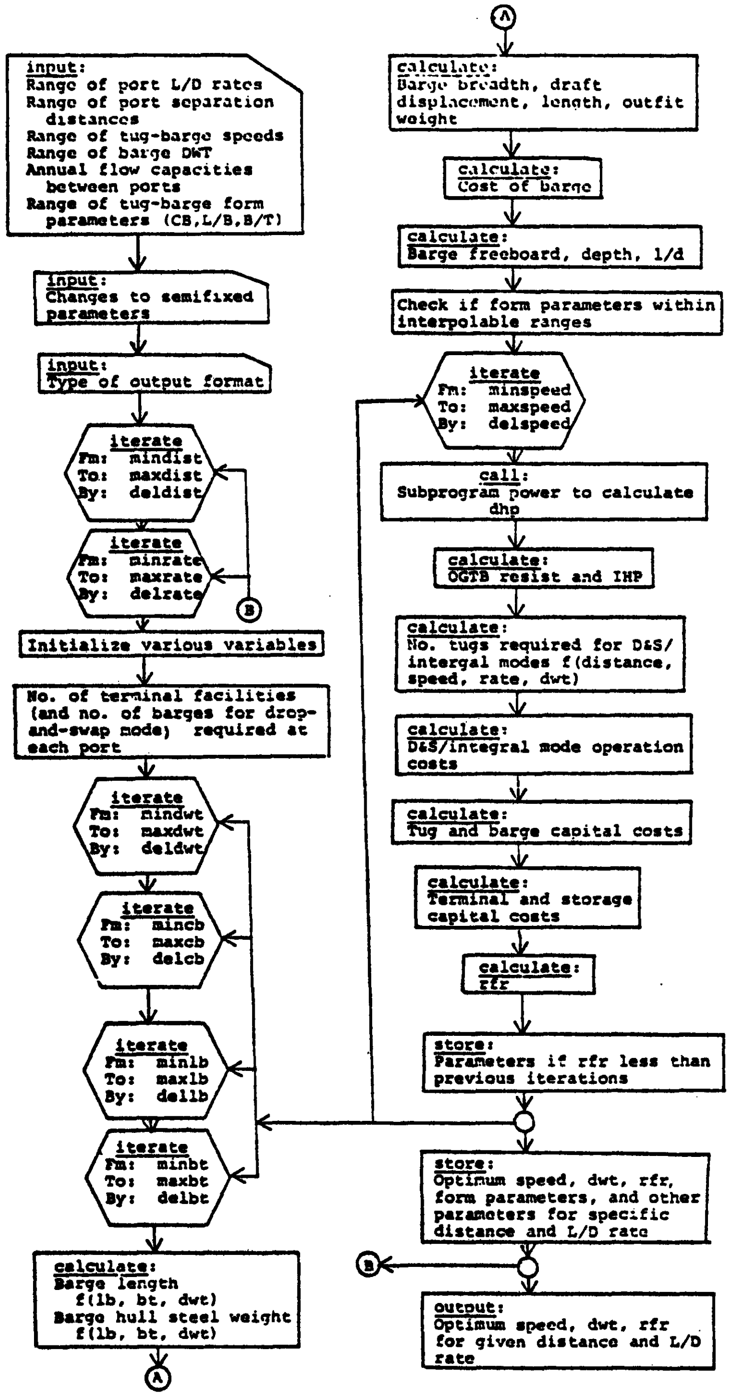


FIGURE 2.3

SUMMARY FLOWCHART FOR DROP-AND-SWAP

COMPUTER MODEL

```
drop_and_swap
Input via list format the following parameters:

Do you wish to specify individual port L-D rates?: no

minrate, maxrate, delrate:  2000.,14000.,2000.,

mindist, maxdist, deldist:  500.,5000.,1500.,

minspeed, maxspeed, delspeed:  6.,13.,1.,

mindwt, maxdwt, deldwt:  5000.,100000.,5000.,

aflowave1, aflowave2:  1000000.,0.,

mincb, maxcb, delcb:  .75,.83,.01,

minlb, maxlb, dellb:     6.,6.4,.2,

minbt, maxbt, delbt:    2.,3.,.5,

input changes to semifixed data via get data format   maxt1=38., maxt2=38.;

Do you want printed output?: y

Do you want detailed output?: y
```

FIGURE 2.4

SAMPLE INPUT FOR THE DROP-AND-SWAP MODEL

This is the semi-fixed parametric data file.

tugopdays	350.00
bargeopdays	350.00
tlink (hrs)	4.00
tunlink (hrs)	4.00
servmargin	0.20
fuelmargin	0.20
sfc (lb/HP-hr)	0.36
cfuel ($/LT)	140.00
clube ($/gal)	1.75
nrcrew	16.00
cwages ($/yr)	65000.00
csubs ($/yr)	3500.00
csteelt ($1000/LT)	2.94
coutfitt ($1000/LT)	15.08
csteelb ($1000/LT)	1.10
coutfitb ($1000/LT)	12.82
ltug (ft)	140.00
wmisc (LT)	460.00
aother ($/yr)	30000.00
admin ($/yr)	150000.00
cfixport1 ($/call)	0.00
cfixport2 ($/call)	0.00
cvarport1 ($/DWT)	0.00
cvarport2 ($/DWT)	0.00
cfixterm ($/DWT-day)	0.00
cvarterm ($/DWT)	0.00
cfixstor ($/DWT)	0.00
cvarstor ($/DWT)	0.00
delay1 (hrs)	4.00
delay2 (hrs)	4.00
maxl (ft)	999.00
maxb (ft)	999.00
maxt1 (ft)	99.00
maxt2 (ft)	99.00
disrate	0.10
econlife (yrs)	20.00
inflafctr	0.00
vcargo ($/DWT)	200.00

BASE CASE VALUES FOR

SEMI-FIXED PARAMETRIC DATA

FIGURE 2.5

DWT (1000)	SPD (KTS)	RFR ($)	CB	BLENGTH	L/B	B/T	L/D	ACOST ($M)	CTUG ($M)	CBARGE ($M)	IHP	NO_TUG/ITB	NRTRIPS	SEATIMET (DAYS)	PORTIMET (DAYS)	AFLOWCAP	TWAIT1/$ (DAYS)
5	6.0	14.18	0.78	200.9	6.00	2.00	7.0	14.180	5.941	1.194	516	5	44.06	7.94	0.00	220280	0.00
5	6.0	16.53	0.78	200.9	6.00	2.00	7.0	16.531	5.941	1.194	516	6	33.51	7.94	2.50	167553	0.00
5	6.0	20.83	0.78	200.9	6.00	2.00	7.0	20.830	5.941	1.194	516	7	28.51	7.28	5.00	142534	41
5	7.0	11.84	0.78	200.9	6.00	2.00	7.0	11.836	6.085	1.194	800	4	50.34	6.95	0.00	251712	0.00
5	7.0	16.84	0.78	200.9	6.00	2.00	7.0	16.836	6.085	1.194	800	6	37.03	6.95	2.50	185139	0.00
5	7.0	21.18	0.73	200.9	6.00	2.00	7.0	21.178	6.085	1.194	800	7	31.01	6.29	5.00	155063	44
5	8.0	12.18	0.78	200.9	6.00	2.00	7.0	12.177	6.281	1.194	1186	4	56.38	6.21	0.00	281879	0.00
5	8.0	14.52	0.78	200.9	6.00	2.00	7.0	14.520	6.281	1.194	1186	5	40.19	6.21	2.50	200957	0.00
5	8.0	21.64	0.78	200.9	6.00	2.00	7.0	21.638	6.281	1.194	1186	7	33.20	5.54	5.00	166008	47
5	9.0	12.56	0.78	200.9	6.00	2.00	7.0	12.558	6.498	1.194	1610	4	62.17	5.63	0.00	310855	0.00
5	9.0	14.89	0.78	200.9	6.00	2.00	7.0	14.890	6.498	1.194	1610	5	43.05	5.63	2.50	215262	0.00
5	9.0	18.36	0.78	200.9	6.00	2.00	7.0	18.360	6.498	1.194	1610	6	35.13	4.96	5.00	175651	50
5	10.0	10.47	0.78	200.9	6.00	2.00	7.0	10.471	7.056	1.194	2685	3	67.74	5.17	0.00	338710	0.00
5	10.0	15.84	0.78	200.9	6.00	2.00	7.0	15.845	7.056	1.194	2685	5	45.65	5.17	2.50	228261	0.00
5	10.0	19.38	0.78	200.9	6.00	2.00	7.0	19.377	7.056	1.194	2685	6	36.84	4.50	5.00	184211	53
5	11.0	11.08	0.78	200.9	6.00	2.00	7.0	11.083	7.518	1.194	3560	3	73.10	4.79	0.00	365506	0.00
5	11.0	16.60	0.78	200.9	6.00	2.00	7.0	16.598	7.518	1.194	3560	5	48.02	4.79	2.50	240125	0.00
5	11.0	20.17	0.78	200.9	6.00	2.00	7.0	20.169	7.518	1.194	3560	6	38.37	4.12	5.00	191860	55
5	12.0	14.12	0.78	200.9	6.00	2.00	7.0	14.122	9.858	1.194	7799	3	78.26	4.47	0.00	391304	0.00
5	12.0	16.48	0.78	200.9	6.00	2.00	7.0	16.485	9.858	1.194	7799	4	50.20	4.47	2.50	250996	0.00
5	12.0	24.17	0.78	200.9	6.00	2.00	7.0	24.172	9.858	1.194	7799	6	39.75	3.81	5.00	198738	57
5	13.0	15.68	0.78	200.9	6.00	2.00	7.0	15.676	11.106	1.194	9942	3	83.23	4.21	0.00	416159	0.00
5	13.0	17.97	0.78	200.9	6.00	2.00	7.0	17.971	11.106	1.194	9942	4	52.20	4.21	2.50	260994	0.00
5	13.0	21.21	0.78	200.9	6.00	2.00	7.0	21.209	11.106	1.194	9942	5	40.99	3.54	5.00	204955	59

DWT (1000)	SPD (KTS)	RFR ($)	CB	BLENGTH	L/B	B/T	L/D	ACOST ($M)	CTUG ($M)	CBARGE ($M)	IHP	NO_TUG/ITB	NRTRIPS	SEATIMET (DAYS)	PORTIMET (DAYS)	AFLOWCAP	TWAIT1/$ (DAYS)
5	6.0	14.30	0.78	222.6	6.00	2.50	8.9	14.300	5.934	1.315	502	5	44.06	7.94	0.00	220280	0.00
5	6.0	16.63	0.78	222.6	6.00	2.50	8.9	16.630	5.934	1.315	502	6	33.51	7.94	2.50	167553	0.00
5	6.0	20.94	0.78	222.6	6.00	2.50	8.9	20.940	5.934	1.315	502	7	28.51	7.28	5.00	142534	41
5	7.0	11.98	0.78	222.6	6.00	2.50	8.9	11.980	6.095	1.315	821	4	50.34	6.95	0.00	251712	0.00
5	7.0	16.98	0.78	222.6	6.00	2.50	8.9	16.976	6.095	1.315	821	6	37.03	6.95	2.50	185139	0.00
5	7.0	21.33	0.78	222.6	6.00	2.50	8.9	21.333	6.095	1.315	821	7	31.01	6.29	5.00	155063	44
5	8.0	12.37	0.78	222.6	6.00	2.50	8.9	12.373	6.320	1.315	1262	4	56.38	6.21	0.00	281879	0.00
5	8.0	14.70	0.78	222.6	6.00	2.50	8.9	14.695	6.320	1.315	1262	5	40.19	6.21	2.50	200957	0.00
5	8.0	21.86	0.78	222.6	6.00	2.50	8.9	21.861	6.320	1.315	1262	7	33.20	5.54	5.00	166008	47
5	9.0	12.98	0.78	222.6	6.00	2.50	8.9	12.976	6.659	1.315	1923	4	62.17	5.63	0.00	310855	0.00
5	9.0	15.28	0.78	222.6	6.00	2.50	8.9	15.282	6.659	1.315	1923	5	43.05	5.63	2.50	215262	0.00
5	9.0	18.77	0.78	222.6	6.00	2.50	8.9	18.771	6.659	1.315	1923	6	35.13	4.96	5.00	175651	50
5	10.0	10.54	0.78	222.6	6.00	2.50	8.9	10.541	7.025	1.315	2626	3	67.74	5.17	0.00	338710	0.00
5	10.0	15.69	0.78	222.6	6.00	2.50	8.9	15.895	7.025	1.315	2626	5	45.65	5.17	2.50	228261	0.00
5	10.0	19.42	0.78	222.6	6.00	2.50	8.9	19.420	7.025	1.315	2626	6	36.84	4.50	5.00	184211	53
5	11.0	11.53	0.78	222.6	6.00	2.50	8.9	11.526	7.766	1.315	4025	3	73.10	4.79	0.00	365506	0.00
5	11.0	17.12	0.78	222.6	6.00	2.50	8.9	17.122	7.766	1.315	4025	5	48.02	4.79	2.50	240125	0.00
5	11.0	20.72	0.78	222.6	6.00	2.50	8.9	20.717	7.766	1.315	4025	6	38.37	4.12	5.00	191860	55
5	12.0	12.37	0.78	222.6	6.00	2.50	8.9	12.375	8.416	1.315	5222	3	78.26	4.47	0.00	391304	0.00
5	12.0	14.70	0.78	222.6	6.00	2.50	8.9	14.703	8.416	1.315	5222	4	50.20	4.47	2.50	250996	0.00
5	12.0	21.79	0.78	222.6	6.00	2.50	8.9	21.788	8.416	1.315	5222	6	39.75	3.81	5.00	198738	57
5	13.0	17.00	0.78	222.6	6.00	2.50	8.9	17.004	12.100	1.315	11599	3	83.23	4.21	0.00	416159	0.00
5	13.0	19.28	0.78	222.6	6.00	2.50	8.9	19.278	12.100	1.315	11599	4	52.20	4.21	2.50	260994	0.00
5	13.0	22.59	0.78	222.6	6.00	2.50	8.9	22.588	12.100	1.315	11599	5	40.99	3.54	5.00	204955	59

DWT (1000)	SPD (KTS)	RFR ($)	CB	BLENGTH	L/B	B/T	L/D	ACOST ($M)	CTUG ($M)	CBARGE ($M)	IHP	NO_TUG/ITB	NRTRIPS	SEATIMET (DAYS)	PORTIMET (DAYS)	AFLOWCAP	TWAIT1/$ (DAYS)
5	6.0	14.92	0.78	242.4	6.00	3.00	10.8	14.924	5.967	1.798	568	5	44.06	7.94	0.00	220280	0.00
5	6.0	17.17	0.78	242.4	6.00	3.00	10.8	17.172	5.967	1.798	568	6	33.51	7.94	2.50	167553	0.00
5	6.0	21.54	0.78	242.4	6.00	3.00	10.8	21.540	5.967	1.798	568	7	28.51	7.28	5.00	142534	41
5	7.0	12.58	0.78	242.4	6.00	3.00	10.8	12.579	6.150	1.798	929	4	50.34	6.95	0.00	251712	0.00

FIGURE 2.6

EXAMPLE OF DETAILED OUTPUT FROM THE DROP-AND-SWAP COMPUTER MODEL

```
aflowave1  =  1000000   minrate =        2000   mindist =       500   minspeed =      6.00    mindwt  =       5000
aflowave2  =        0   maxrate =       14000   maxdist =      5000   maxspeed =     13.00    maxdwt  =     100000
                        delrate =        2000   deldist =      1500   delspeed =      1.00    deldwt  =       5000

maxi  =        999.00   tugopdays    =    350   servmargin =   0.20   csteelt  =      2.94    csteelb    =    1.10
maxb  =        999.00   bargeopdays  =    350   fuelmargin =   0.20   coutfitt =     15.08    coutfitb   =   12.82
maxt1 =         38.00   tlink        =      4   sfc        =   0.36   cwages   =  65000.00    cfixport1  =    0.00
maxt2 =         38.00   tunlink      =      4   cfuel      = 140.00   csubs    =   3500.00    cfixport2  =    0.00
itug  =        140.00   delay1       =      4   clube      =   1.75   nrcrew   =     16.00    cvarport1  =    0.00
                        delay2       =      4   vcargo =     100.00                           cvarport2  =    0.00

actner =     30000.00   disrate   =      0.10   mincb =   0.75   minlb =   6.0   minbt =   2.0   cvarstor  =    0.00
admin  =    150000.00   econlife  =     20.00   maxcb =   0.83   maxlb =   6.4   maxbt =   3.0   cfixstor  =    0.00
                        pvf       =      8.51   delcb =   0.01   dellb =   0.2   delbt =   0.5   cvarterm  =    0.00
                        inflafctr=       0.00                                                    cfixterm  =    0.00

DWT   SPD   RFR    CB  BLENGTH  L/B  B/T  L/D   ACOST   CTUG   CBARGE  IHP NO_TUG/ITB NRTRIPS SEATIMET PORTIMET AFLOWCAP TWAIT1/2
(1000)(KTS) ($)                                 ($M)    ($M)   ($M)                            (DAYS)   (DAYS)            (DAYS)

rate =     2000
  distance =       500  minbarge1 =          2  minbarge2 =      2
 15  10.0    6.85 0.78 316.9   6.00 2.00  7.6   6.852   7.839   4.294  4160     1        67.74    5.17     0.00  1016129    0.00
 15   8.0   13.84 0.81 312.1   6.00 2.00  7.6  13.838   6.841   4.193  2273     4        17.04    5.54    15.00   255578   73

rate =     4000
  distance =       500  minbarge1 =          1  minbarge2 =      1
 15  10.0    5.65 0.78 316.9   6.00 2.00  7.6   5.645   7.839   4.294  4160     1        67.74    5.17     0.00  1016129    0.00
 25   9.0    8.78 0.78 426.1   6.00 2.50  9.8   8.779   8.009   8.513  4476     2        20.04    4.96    12.50   501060   72

rate =     6000
  distance =       500  minbarge1 =          1  minbarge2 =      1
 15  10.0    5.65 0.78 316.9   6.00 2.00  7.6   5.645   7.839   4.294  4160     1        67.74    5.17     0.00  1016129    0.00
 20   6.0    7.23 0.81 349.8   6.00 2.00  7.7   7.230   6.212   5.611  1051     2        25.10    7.28     6.67   501992   48

rate =     8000
  distance =       500  minbarge1 =          1  minbarge2 =      1
 15  10.0    5.65 0.78 316.9   6.00 2.00  7.6   5.645   7.839   4.294  4160     1        67.74    5.17     0.00  1016129    0.00
 15   7.0    6.95 0.81 312.1   6.00 2.00  7.6   6.952   6.434   4.193  1486     2        34.88    6.29     3.75   523132   37

rate =    10000
  distance =       500  minbarge1 =          1  minbarge2 =      1
 15  10.0    5.65 0.78 316.9   6.00 2.00  7.6   5.645   7.839   4.294  4160     1        67.74    5.17     0.00  1016129    0.00
 15   6.0    6.78 0.81 312.1   6.00 2.00  7.6   6.776   6.143   4.193   914     2        34.05    7.28     3.00   510811   29

rate =    12000
  distance =       500  minbarge1 =          1  minbarge2 =      0
 20  10.0    5.52 0.81 349.8   6.00 2.00  7.7   5.517   8.413   5.611  5219     1        51.22    5.17     1.67  1024390    1.67
 35   7.0    5.07 0.79 508.4   6.40 2.50 10.7   5.072   7.035  12.403  2645     1        28.88    6.29     5.83  1010806   48

rate =    14000
  distance =       500  minbarge1 =          1  minbarge2 =      0
 20  10.0    5.54 0.81 349.8   6.00 2.00  7.7   5.536   8.413   5.611  5219     1        53.07    5.17     1.43  1061372    1.43
 20  11.0    4.78 0.78 355.1   6.00 2.00  7.7   4.777   9.149   5.724  6546     1        50.16    4.12     2.86  1003102   41
```

FIGURE 2.7

EXAMPLE OF SUMMARY OUTPUT FROM DROP-AND-SWAP MODEL

$$\text{Barge Freeboard (inches)} = \text{"fbd"} = 0.75 \times \Big[\underbrace{\left(\begin{matrix}\text{Minimum Freeboard Value}\\ \text{Found in Table 2.5}\end{matrix}\right) \times \left(\frac{\text{"cb"}+0.68}{1.38}\right)}_{\text{"minfbd"}} + \text{"shrfact"} + \text{"ldfact"} \Big]$$

Where: 0.75 accounts for the 25% freeboard reduction pertaining to unmanned barges

"shrfact"= sheer factor correction $= 0.0375 \times \text{"lbarge"} + 3.75$

$$\text{"ldfact"} = \begin{matrix}\text{length-depth ratio}\\ \text{correction}\end{matrix} = \begin{cases} = 0 & \text{if } \dfrac{\text{"lbarge"}}{\text{depth}} > 15 \\[2ex] = \left(\text{depth} - \dfrac{\text{"lbarge"}}{15}\right) \times \left[\dfrac{\dfrac{\text{"lbarge"}}{132.2}}{1 - \dfrac{\text{"lbarge"}}{12 \times 131.2}}\right] & \text{if } \begin{cases} \dfrac{\text{"lbarge"}}{\text{depth}} < 15 \\ \text{"lbarge"} < 393.6' \end{cases} \\[4ex] = 4 \times \left(\text{depth} - \dfrac{\text{"lbarge"}}{15}\right) & \text{if } \begin{cases} \dfrac{\text{"lbarge"}}{\text{depth}} < 15 \\ \text{"lbarge"} > 393.6' \end{cases} \end{cases}$$

$$\text{where depth} = \left(\frac{\text{"minfbd"}}{12} + \text{"tbarge"}\right)$$

SOURCE: Chapter III of the IMCO International Conference on Load Lines, 1966

FIGURE 2.8

EQUATIONS USED IN THE CALCULATION OF BARGE FREEBOARD

TABLE 2.1

RANGES FOR SYSTEM PARAMETERS

Parameter	Lower Bound	Upper Bound	Restrictions/ Comments	Reason for Bound
Barge DWT	9,000 6,000	80,000 100,000	 Reduced confidence range	Sharp (1975) Valid ranges for operating cost formulae
Barge Length	100'	750'		ABS (1973) Valid ranges for barge hull weight formula
Froude Number	0	0.22	$F_n = \frac{0.298 \text{ Tug Speed}}{\sqrt{\text{Two-Barge Length}}}$	Tsuchida (1969) Valid range for residual resistance coefficients
Barge C_B*	0.775 0.75	0.835 0.85	 Reduced confidence range	" " "
Tug-Barge L/B*	6.2 6.0	7.6 8.0	 Reduced confidence range	" " "
Barge B/T	2.46 2.00	2.76 3.25	 Reduced confidence range	" " "
Barge L/D	0	16.0		ABS (1973) valid range for barge hull weight formula
Tug IHP	5000 3000	35,000 35,000	 Reduced confidence range	Sharp (1975) Valid range for operating cost formulae and tug capital cost formula

*Either C_B or L/B can be in the reduced confidence range, but not both.

TABLE 2.2

FORMULAE FOR DETERMINING NUMBER OF TERMINAL FACILITIES/BARGES AT EACH PORT

Port	Special Conditions	Formulae
1	none	$\text{"minbarge1"} = \text{ceil}\left(\frac{\text{"mflowave1"}}{30.5\text{x"rload1"}} + \frac{\text{"mflowave2"}}{30.5\text{x"runload1"}}\right)$
	"rload1"=0	$\text{"minbarge1"} = \text{ceil}\left(\frac{\text{"mflowave2"}}{30.5\text{x"runload1"}}\right)$
	"runload1"=0	$\text{"minbarge1"} = \text{ceil}\left(\frac{\text{"mflowave1"}}{30.5\text{x"rload1"}}\right)$
2	none	$\text{"minbarge2"} = \text{ceil}\left(\frac{\text{"mflowave1"}}{30.5\text{x"runload2"}} + \frac{\text{"mflowave2"}}{30.5\text{x"rload2"}}\right)$
	"rload2"=0	$\text{"minbarge2"} = \text{ceil}\left(\frac{\text{"mflowave1"}}{30.5\text{x"runload2"}}\right)$
	"runload2"=0	$\text{"minbarge2"} = \text{ceil}\left(\frac{\text{"mflowave2"}}{30.5\text{x"rload2"}}\right)$

Note: The function ceil in the above formulae is used to round to the next largest integer the expression in parenthesis.

TABLE 2.3

BARGE HULL WEIGHT AND LENGTH FORMULAE

Parameter	Block Co-efficient	Formula
Barge Length "lbarge"	0.75	"lbarge75"=$e^{1.128}\ lb^{0.742}\ bt^{0.382}\ dwt^{0.336}$
	0.80	"lbarge80"=$e^{1.108}\ lb^{0.739}\ bt^{0.381}\ dwt^{0.336}$
	0.85	"lbarge85"=$e^{1.088}\ lb^{0.737}\ bt^{0.379}\ dwt^{0.336}$
Barge Hull Weight	0.75	"hullwt75"=$e^{-6.206}lb^{1.348}\ bt^{0.884}\ dwt^{1.104}$
	0.80	"hullwt80"=$e^{-6.396}lb^{1.368}\ bt^{0.909}\ dwt^{1.111}$
"wsteelb"	0.85	"hullwt85"=$e^{-6.569}lb^{1.391}\ bt^{0.930}\ dwt^{1.117}$

TABLE 2.4

BARGE OUTFIT WEIGHT

Formula: "woutfitb" = max(50, 1.496x"lbarge"-284.24)

Barge Length	Outfit Weight
200	50.00
250	89.76
300	164.56
350	239.36
400	314.16
450	388.96
500	463.76
550	538.56
600	613.36
650	688.16
700	762.96
750	837.76

TABLE 2.5

MINIMUM FREEBOARD VALUES

Barge Length	Minimum Freeboard	Barge Length	Minimum Freeboard	Barge Length	Minimum Freeboard
100'	9.8"	320'	43.2"	540'	86.3"
110	10.8	330	45.0	550	88.0
120	11.9	340	46.9	560	89.6
130	13.0	350	48.8	570	91.1
140	14.2	360	50.7	580	92.6
150	15.5	370	52.7	590	94.1
160	16.9	380	54.7	600	95.5
170	18.3	390	56.8	610	96.9
180	19.8	400	58.8	620	98.3
190	21.3	410	60.9	630	99.6
200	22.9	420	62.9	640	100.9
210	24.5	430	65.0	650	102.1
220	26.2	440	67.0	660	103.3
230	27.8	450	69.1	670	104.4
240	29.5	460	71.1	680	105.5
250	31.1	470	73.1	690	106.6
260	32.8	480	75.1	700	107.7
270	34.6	490	77.1	710	108.7
280	36.3	500	79.0	720	109.7
290	38.0	510	80.9	730	110.7
300	39.7	520	82.7	740	111.7
310	41.4	530	84.5	750	112.6

SOURCE: Table A, Chapter III, IMCO International Convention on Load Lines, 1966.

TABLE 2.6

TUG CAPITAL COSTS

$$\text{Tug Hull Weight} = \text{"wsteelt"} = 0.64\left(\frac{IHP}{1000}\right)^2 + 16.79\left(\frac{IHP}{1000}\right) + 378$$

$$\text{Tug outfit Weight} = \text{"woutfitt"} = 0.1866\left(\frac{IHP}{1000}\right)^2 + 2.733\left(\frac{IHP}{1000}\right) + 154$$

$$\text{Tug Machinery Weight} = \text{"wmacht"} = -0.08889\left(\frac{IHP}{1000}\right)^2 + 32.88\left(\frac{IHP}{1000}\right) + 4.999$$

$$\text{Tug Machinery Cost} = 1{,}300^*(0.314 \times IHP + 1730)$$

$$\text{Tug Cost} = 1{,}000\left[2.94^*\left(\text{Tug Hull Weight}\right) + 15.08^*\left(\text{Tug Outfit Weight}\right) + \left(\text{Tug Machinery Cost}\right)\right]$$

IHP	Tug Hull Weight (LT)	Tug Outfit Weight (LT)	Tug Machinery Weight (LT)	Tug Machinery Cost ($1000)	Tug Cost ($1000)
2,000	414.14	160.21	70.40	3065.4	6,698.1
2,500	423.98	162.00	86.64	3269.5	6,958.1
3,000	434.13	163.88	102.84	3473.6	7,220.4
4,000	455.40	167.92	135.10	3881.8	7,752.0
5,000	477.95	172.33	167.18	4290.0	8,293.0
6,000	501.78	177.12	199.08	4698.2	8.843.3
7,000	526.89	182.27	230.80	5106.4	9,403.1
8,000	553.28	187.81	262.35	5515.6	9,972.3
9,000	580.95	193.71	293.72	5922.8	10,550.8
10,000	609.90	199.99	324.91	6331.0	11,138.7
12,000	671.64	213.67	386.76	7147.4	12,342.8
14,000	738.50	228.84	447.90	7963.8	13,584.4
16,000	810.48	245.50	508.32	8780.2	14,863.5
18,000	887.58	263.65	568.04	9596.6	16,180.2
20,000	969.80	283.30	627.04	10413.0	17,534.4
22,500	1079.78	309.96	699.80	11433.5	19.280.1
25,000	1197.75	338.95	771.44	12454.0	21,084.4
27,500	1323.73	370.27	841.98	13474.5	22,947.3
30,000	1457.70	403.93	911.40	14495.0	24,869.0

SOURCE: Sharp (1975)

*Cost factors inflated 30% from values given in source.

TABLE 2.7

ANNUAL OPERATING COSTS FOR SUPPLIES AND EQUIPMENT, MAINTENANCE AND REPAIR, AND INSURANCE

Supplies and Equipment = "asupplies" = 1.3*(40+0.0018xIHP + 0.00026 x TDWT)

Maintenance and Repair = "amandr" $= 1.5^*(128.8 + 4.539(\frac{IHP}{1000}) - 0.04(\frac{IHP}{1000})^2 + 2.477(\frac{TDWT}{1000}) - 0.009107(\frac{TDWT}{1000})^2$

Insurance = "ainsur" = 1.5*(210 + 0.0036 x IHP + 0.0018 x TDWT)

TDWT	IHP	Supplies & Equipment ($1000)	Maintenance & Repair ($1000)	Insurance ($1000)
5,000	2,000	58.4	224.8	339.3
"	6,000	67.7	250.1	360.9
"	10,000	77.1	273.5	382.5
"	14,000	86.5	295.0	404.1
20,000	2,000	63.4	275.4	379.8
"	6,000	72.8	300.7	401.4
"	10,000	82.2	324.1	423.0
"	14,000	91.5	345.6	444.6
35,000	2,000	68.5	319.9	420.3
"	6,000	77.9	345.2	441.9
"	10,000	87.2	368.6	463.5
"	14,000	96.6	390.1	485.1
50,000	2,000	73.6	358.2	460.8
"	6,000	82.9	383.5	482.4
"	10,000	92.3	406.9	504.0
"	14,000	101.7	428.4	525.6
"	18,000	111.0	447.9	547.2
65,000	2,000	78.7	390.4	501.3
"	6,000	88.0	415.7	522.9
"	10,000	97.4	439.1	544.5
"	14,000	106.7	460.6	566.1
"	18,000	116.1	480.1	587.7

SOURCE: Sharp (1975)
NOTE: TDWT=Cargo Deadweight + Fuel Weight +Miscellaneous Weight
*Inflation Factor to bring cost to 1 Jan 79 level

TABLE 2.8

COMPARISON OF MODEL AND MARAD DATA
FOR ANNUAL OPERATING COSTS

Vessel Type	Annual Crew Wages	Annual Subsistence Costs	Annual M&R Costs	Annual Supplies & Equipment	Annual Insurance	Annual M&R Supplies & Equipment	Other Annual Costs	Total Operating Costs
5,000 DWT 3,000 IHP cy 1978	658,152	23,725	200,000	63,270	496,000	759,270	36,000	1,477,147
Model 1 Jan 1979	845,000	45,500	231,321	60,710	344,700	636,731	30,000	1,557,231
33,500 DWT 15,000 IHP 1 Jan 77	821,250	47,815	405,150	127,385	542,755	1,075,290	29,200	1,973,555
Model 1 Jan 1979	1,040,000	56,000	390,966	98,423	486,450	975,839	30,000	2,101,839
40,000DWT 18,000 IHP cy 1980	1,060,000	50,000	460,683	71,317	352,250	884,250	189,600	2,183,850
Model 1 Jan 1979	1,040,000	56,000	442,247	107,640	520,200	1,070,387	180,000	2,346,387
40,000 DWT 18,000 IHP cy 1980	1,100,000	60,000	705,000	220,000	561,000	1,486,000	20,000	2,666,000
Model 1 Jan 1979	1,040,000	56,000	442,747	107,640	520,200	1,070,387	30,000	2,196,387

SOURCE: Data Received from Maritime Administration Office of Ship Construction, December 1978.

CHAPTER 3

BARGE DESIGN PROGRAM TO ESTIMATE HULL STEEL WEIGHT

3.1 Introduction

The barge design program has been developed to provide an estimate of the hull weight of tank barges as a function of their size and form. This weight estimate is used by the drop-and-swap program to calculate the barge capital construction costs. This is done by multiplying the hull steel weight estimate by a dollar per ton factor to yield the hull cost which is added to the outfit cost to yield the barge cost estimate. This approach was taken rather than a simple regression equation through historical costs because too few large tank barges have been built to yield a reliable prediction equation, and those that have been built were of various designs and arrangements so that their costs were really not comparable. Also, since barge capital costs make up a significant part of the total cost of a tug-barge project, it was deemed necessary to develop a non-trivial model for obtaining reasonably accurate and consistent estimates of the hull steel weight. An additional benefit derived from the barge design model is that it provides hull weights for various barge forms

(length-breadth ratio, breadth-draft ratio, and block coefficient) for a given deadweight. This information is used by the drop-and-swap program to determine the least costly form with respect to capital and propulsive costs.

The barge design model's approach for estimating the weight of a barge of given size, form, and longitudinal frame and transverse girder spacing is basically the same as used by a naval architect in his preliminary design work. That is, the program determines the minimum weight mid-ship scantlings that will satisfy the requirements promulgated in the American Bureau of Shipping's offshore barge (ABS (1973)) and steel vessel (ABS (1978)) rules and that will conform with usual shipyard practice. From the midship scantlings a weight per longitudinal foot can be determined from which the hull weight can be estimated. The program does optimize in the sense that it selects the longitudinal frame and transverse girder spacings within user specified ranges and increments that yield the minimum weight scantlings. It was felt that this parametric optimization procedure which determines a local minimum within the specified parametric ranges was the simplest though not most elegant approach to take. A more theoretical optimization procedure using a stress and bending moment analysis could probably be used to develop a more rational barge design model, but it most likely could not take account of the complexities involved in classification society rules that are used in barges constructed today.

It should be mentioned that it has been assumed that the minimum weight barge would be the optimum. In reality, the minimum weight design may not be the least costly. For example, a design with wider longitudinal spacing may be less expensive due to reduced welding and cutting costs than the less heavy design of narrower spacing. However, it was found from the model's outputs that the calculated hull weight was not very sensitive to longitudinal spacing; (1) so that as long as a reasonable minimum longitudinal spacing is specified (eliminating the possibility of unreasonably small and numerous longitudinal frames), the minimum hull weight design should not be very far from the optimal, minimum cost design.

The design model was originally developed for the single-skin, center longitudinal tank barge design shown in Figure 3.1 for the following two reasons: (1) these barges have the simplest scantling form and thus were the easiest to model, and (2) these types of barges have been the most prevalent of the large ocean going barges so that actual barge hull weight data could be obtained to calibrate the model. However, if proposed U. S. Coast Guard anti-pollution regulations become effective, no single-skin tank barges will be constructed in the future. Tank barges will be required either to have double hulls or protective segregated ballast

---

(1) As an example, the minimum hull steel weights calculated for the six longitudinal spacings (1.50'-2.75') shown in Figure 3.9 differ less than 3%.

tanks located either in double sides, bottoms, or hulls. If this change occurs, the single-skin design model will have to be modified to account for the additional internal structure. In order to see how just one of these types of modifications will affect the hull weight of a barge of specified deadweight, the single-skin model was written with the option for calculating the hull weights of barges with double-bottom scantlings as shown in Figure 3.2. The comparison of single-skin and double-bottom tank barges presented in Section 3.6 indicates the approximate additional weight (and cost) such internal structure will cause.

In the text that follows, the logical structure of the single-skin tank barge program and the modifications required to allow for double-bottoms are presented. This is followed by a presentation and discussion of the results of the program.

3.2 Logical Structure

The logical structure of the program is shown in flow chart form in Figure 3.3. In the text following, each block of the flow chart will be expanded so that the equations (with associated references) and assumptions used will be enumerated. It should be mentioned that it has been assumed for simplicity that deck, bottom, side, and bulkhead longitudinal frame spacings will be the same. The same holds for transverse girder spacings.

3.2.1 Input of Parameter Values

The program requires as input the barge principal dimensions, block coefficient, and scantling arrangement option (single-skin or double-bottom), as well as the parametric range and increment for the spacing of longitudinal frames and transverse girders. The ranges and increments chosen should conform with current shipbuilding practice. Specifically, the following values for the barge design must be provided:

"l":	scantling length at the waterline
"b":	molded breadth
"d":	molded depth
"t":	summer waterline draft
"cb":	block coefficient
"minlonspac":	minimum longitudinal frame spacing to be investigated
"maxlonspac":	maximum longitudinal frame spacing to be investigated
"minxvrspac":	minimum transverse girder spacing to be investigated
"maxxvrspac":	maximum transverse girder spacing to be investigated
"incrlonspac":	the increment to be used in the parametric variation of longitudinal frame spacing from its minimum to its maximum allowed value
"incrxvrspac":	the increment to be used in the parametric variation of transverse web spacing from its minimum value to its maximum allowed value

Limitations on the values of these parameters as given within the specified paragraphs of ABS (1973) are as follows:

Limitation	Paragraph
100 ft $\leq$ "l" $\leq$ 750 ft	(Table 3.2)
"l"/"d" < 15	(3.5)
"maxlonspac" < 3 1/3 ft	(5.1.1)

3.2.2 Calculation of Required Section Modulus

In accordance with ABS (1973, par. 3.3.1), the program determines the required hull-girder section modulus at amidships ("smreq") as follows:

"smreq" = "fact" x "b" x ("cb" + 1/2)

where "fact" has its value determined by its length from ABS (1973, Table 3.2) which is replicated in Table 3.1. This section modulus is the best approximation that can be made for design purposes without knowing the actual bending moments that the barge would experience in still water.

3.2.3 Calculation of the Unsupported Spans of Transverse Girders and Required Number of S tanchions

The unsupported span of deck and bottom transverse girders is shown as "l2" in Figures 3.1 and 3.2. "l2" has

been limited to less than 15 feet in accordance with ABS (1978, para. 11.5) by inserting stanchions equally spaced between the bulkhead and side girders. To obtain the required span and the number of stanchions ("stans"), the program assumes certain arbitrary values for the depth of longitudinal bulkhead and side girders as shown in Table 3.2. These assumptions, although arbitrary, have been shown reasonable when compared with actual barge scantlings.

The unsupported span of longitudinal bulkhead and side girders is shown as "l4" in Figure 3.1. It is essentially the length between the top of the bottom girder (or double bottom plate) to the bottom of the deck girder.

3.2.4 Beginning of Program Iterations with respect to Transverse Girder and Longitudinal Frame Spacing

At this point the program has finished its initialization and begins to calculate values with respect to a specific longitudinal frame and transverse girder spacing. The first thing that is calculated is the spacing of transverse bulkheads for tanks. In this calculation, it has been assumed that tank length will be limited to less than 15% of the barge length in accordance with MARPOL '73, that tanks will be of equal length, and that the minimum integral number of transverse girders are to be included in each tank.

Following this calculation, the program begins the main task of calculating the scantlings required to fulfill the minimum requirements specifed in the ABS (1973) and (1978).

Scantlings are calculated for bottom members, followed by side members, deck members, and finally the longitudinal bulkhead members. The actual scantlings are calculated by the use of two subroutines--"smact1" for longitudinal frames and "smact2" for transverse girders.

The subroutine "smact1" requires the following inputs:

1. The thickness of the shell plating
2. The minimum required section modulus for the member
3. The effective width of shell plating to be used in the section modulus calculations within the subroutine.

The subroutine "smact2" requires similar inputs, except for the addition of the depth of the associated longitudinal frame which is obtained as output from subroutine "smact1".

3.2.5 Derivation of the Inputs for Subroutines "smact1" and "smact2"

3.2.5.1 Plate thickness. The thicknesses of plating to be used as inputs into subroutines "smact1" and "smact2" are the minimums allowed by ABS (1973). The appropriate rules are summarized in Table 3.3. These thicknesses are rounded up to the nearest 1/16 inch in order to conform with standard plate stock. For the same reasoning, plate stock is limited to 1.4375 inches.

3.2.5.2 Section modulus. The section moduli to be used as inputs into subroutines "smact1" and "smact2" are the minimums required by ABS (1973). The appropriate rules, including the definition of pressure heads, are summarized in

Table 3.4 for longitudinal frames and in Table 3.5 for transverse girders.

3.2.5.3 Effective plating. The width of effective plating to be used in subroutines "smact1" and "smact2" is the minimum of the following three values based on the referenced rules:

1. Longitudinal frame spacing ("lonspac") for subroutine "smact1"; transverse girder spacing ("xvrspac") for subroutine "smact2", per ABS (1973, par. 1.25.1)
2. One third the unsupported span of the member (l in Tables 3.4 and 3.5), per ABS (1973, par. 1.25.1)
3. Sixty times the plate thickness as per Evans (1975)

3.2.6 Calculation of Hull Girder Section Modulus

At this point, the minimum scantlings required to satisfy the individual frames and girders have been determined by repetitive calling of the subroutines "smact1" and "smact2". The next step is to determine whether the individual longitudinal members (plates and longitudinal frames) provide sufficient material to satisfy the minimum hull-girder section modulus requirement calculated in Section 3.2.2. This is done by using the method shown in Comstock (1967, pp.183-184) which uses all longitudinal material in the calculation. In our case, this includes bottom, side, deck and longitudinal bulkhead longitudinal frames and plates.

3.2.7 Strategy to Satisfy Minimum Hull Girder Section Modulus Requirement

If the calculation of the bottom and deck hull-girder section modulus results in values greater than the minimum required, then the hull weight is calculated, as explained in the Section 3.2.8. Otherwise, it is necessary to increase the dimensions of certain of the longitudinal members in such a way that should provide the maximum increase in the calculated deck and/or bottom section moduli with the smallest amount of additional material. Since the barge hull is essentially a stiffened box girder, is is apparent that material should be added to the extreme flanges--the deck and bottom plates and longitudinals--to get the maximum increase in section modulus for the given material added. This is exactly what the program does. It begins with the deck longitudinal members since the deck section modulus will be initially smaller than the bottom section modulus due to smaller hydrostatic pressure experienced by the deck members.

Specifically, the program increases the dimensions of deck scantlings in the order: (1) deckplate (by 1/16"s), (2) deck longitudinal flange width (by 1/2"s), and (3) deck longitudinal flange thickness (by 1/16"s). The program increases the first dimension on the list that is not constrained by usual yard practice or structural considerations. This procedure continues until the deck modulus is at least as great as the bottom modulus. Then, if

the deck and bottom section moduli are still smaller than the required section modulus, the program will increase the dimensions of the bottom scantlings in the order: (1) bottom plate (by 1/16"s), (2) bottom longitudinal flange width (by 1/2"s), and (3) bottom longitudinal flange thickness (by 1/16"s). The program increases the first dimension on the list that is not constrained by yard practice or structural considerations. This procedure continues until the bottom section modulus exceeds the deck section modulus; then the program goes back to increase the deck scantlings. This iterative procedure keeps the deck and bottom section moduli balanced until both exceed the required hull-girder section modulus, or until all constraints become binding. In this case, the program prints an error message that states the scantlings cannot be increased sufficiently to provide the required section modulus. The program then continues by incrementing the spacing of either the longitudinals or transverses and proceeding with the individual section moduli calculations.

The constraints used in this iterative procedure are as follows:

1. Web depth > 2 x flange width
2. Plate thickness < 1 7/16"
3. Flange thickness < (web thickness + 1/8")
4. Flange width < 32 x web thickness

The first three constraints are based on reasonable yard

practices, and the last is used to conform with Jenkins (1977) to ensure flange material is effectively utilized.

The hull-girder section modulus routine has two special features that help ensure longitudinal scantlings are of reasonable dimensions and minimum weight. The first is that when either the deck or bottom plate is increased in thickness, the program calls the subroutine "smact1" to ensure the associated longitudinal stiffener has scantlings that reasonably match the plate to which it is attached.

The second feature is one that limits the deck plate thickness to "n"/16 inches greater than the bottom plate before any increases are made to the other two deck scantlings. The value of "n" is initially set to zero and is then increased incrementally. For each value of "n", longitudinal member scantlings are calculated to satisfy the hull section modulus requirement. The resultant hull weight is then determined (as per Section 3.2.8), and "n" is allowed to increase until hull weight no longer decreases. This feature was heuristically determined to provide the minimum hull weight satisfying the required hull-girder section modulus for a specified longitudinal frame and transverse girder spacing.

3.2.8 Calcuation of Hull Weight

Given the longitudinal member scantlings obtained from the previous hull-girder section modulus routine, the program calculates the resulting longitudinal steel cross-sectional

area. From this, the longitudinal weight per foot is determined simply by multiplying the cross-sectional area in square inches by the steel weight factor: 3.4 pounds per square inches-foot. To this must be added the longitudinal weight per foot contributed by bottom, side, deck, and longitudinal bulkhead transverse girders, as well as by the stanchions and transverse bulkhead plates with associated stiffeners. The details of how the weights per foot for these transverse members are determined are shown in Table 3.6. The hull weight is then simply found by multiplying the weight in pounds per longitudinal foot by the factor: "1"/2240, which converts it into weight in long tons. This weight could then be multiplied by a correlation factor which would account for the different scantlings used in the bow and stern areas as well as weld weights. However, at this point, it appears that the hull weight calculated is in reasonable agreement with the hull weights of single-skin tank barges currently in operation.

The program finishes the routine for a specific value of longitudinal and transvers girder spacing and "n" by storing these values and those of the hull weight and associated deck and bottom hull girder section moduli if the calculated hull weight is less than that found during previous iterations. Thus, at the end of all iterations with respect to longitudinal and transverse spacing, the minimum hull weight barge particulars will be available for outputting. It is al-

so at this point of the program that it is determined whether "n" should be incremented. This is done by examining whether the last increase in "n" had changed the calculated values of the deck and bottom hull-girder section moduli. If no change is observed, then iterations with respect to "n" are terminated and either the longitudinal or transverse spacing is incremented.

Finally, at the end of all iterations, the program outputs the values of hull weight, longitudinal and transverse girder spacing, deck and bottom section moduli, and "n" for the barge of the least hull weight found within the parametric ranges and increments of longitudinal and transverse girder spacing.

3.3 Calculation of Longitudinal Frame Scantlings (The Logic of "smact1")

As explained previously "smact1" calculates the scantlings of a longitudinal member given the input:

1. Plate thickness ("t")
2. Plate effective width as a flange ("s")
3. Required section modulus for the member ("smr")

The logical flow of how these scantlings are calculated is shown in Figure 3.4 and will be explained in detail in the following text.

First the subroutine assumes certain initial values for the dimensions of the scantlings. These are:

Dimension	Initial Value
Plate thickness	as specified in inputted value "t"
Plate depth	as specified in inputted value "s"
Web depth	3.5"
Web thickness	plate thickness, for plate thinner than 3/8"
	plate thickness less 1/8", for plate thicker than 3/8"
Flange width	the greater of:
	(1) web thickness
	(2) 3/4"
Flange thickness	plate thickness

These values were chosen as to be reasonable for the smallest allowed scantlings, based on available rolled sections.

The strategy of the subroutine is then to calculate the section modulus of the longitudinal with the above minimum scantlings. If the calculated section modulus is less than that required by the inputted value "smr", the program increments the scantling dimensions in a manner that will attempt to add the least amount of steel area while achieving the required section modulus. It was found from theoretical analysis and empirical observation that the best approach was to increase the dimensions of the scantlings in the order: (1) web depth (by 1/2"s), (2) flange width (by 1/2"s), (3) flange thickness (by 1/16"s), and (4) web thickness (by 1/16"s). The subroutine will increase the first dimension in

the above list that is not constrained by usual yard practice or structural considerations. The constraints based on usual shipyard practice were as follows:

1. 5 x flange width > web depth > 2 x flange width
2. Flange thickness < plate thickness
3. Web thickness < plate thickness
4. Web depth < 18"

Certainly, none of these constraints have to be rigidly maintained, but they do represent reasonable values. The constraints based on strucural considerations were as follows:

1. Web depth less than 75 times web thickness to conform with Horne (1958) to ensure stability against web buckling
2. Flange width less than 32 times web thickness to conform with Jenkins (1977) to ensure flange width material is really effective in the section modulus.

After any scantling is increased, the subroutine recalculates the section modulus. If it exceeds the required section modulus, the subroutine returns the scantling dimensions to the main program. Otherwise, it repeats the above procedure until no dimension can be increased without violating a constraint. At that point an error message prints out that scantling limits have been exceeded in "smact1" and sends an indicator to the main program that this condition exists so appropriate action can be taken.

3.4 Calculation of Transverse Girder Scantlings (The Logic of "smact2")

As explained previously, subroutine "smact2" calculates the scantlings of a transverse girder given the input:

1. Plate thickness ("tp")

2. Plate effective width as a flange ("s")

3. Required section modulus for the member ("smr")

4. Depth of the associated longitudinal frame ("dep"), i.e., depth of the deck longitudinal for a deck girder

The logical flow of how these scantlings are calculated is shown in Figure 3.5 and will be explained in detail in the following text.

First the subroutine assumes some initial values for the dimensions of the scantlings. Many of these initial values are the minimums allowed by ABS (1973) and (1978). The values and the associated references are presented in Table 3.7. The strategy of the program is similar to that described for "smact1". First the subroutine calculates the section modulus of the girder with the above minimum scantlings. If the calculated section modulus is less than that required by the inputed value "smr", then the program increments the scantling dimensions in a manner that will attempt to add the least amount of steel area while achieving the required section modulus. It was found from theoretical analysis and empirical observation of the program that the best approach was to in-

crease the dimensions of the scantlings in the order: (1) web depth (by 1"s), (2) flange width (by 1"s), (3) flange thickness (by 1/16"s), (4) and web thickness (by 1/16"s). The subroutine will increase the first dimension in the list that is not constrained by usual yard practice or structural considerations. The constraints based on usual shipyard practice were:

1. 5 x flange width > web depth > 2 x flange width
2. Web depth less than 48" for bottom and deck girders
3. Web depth less than 80" for side and bulkhead girders
4. Web thickness < (plate thickness + 1/2")
5. Flange thickness < (web thickness + 1/2")
6. Flange thickness < 1 7/16"
7. Web thickness < 1 7/16"

Certainly, none of these constraints have to be rigidly maintained, but they do represent reasonable values as verified by conversations with shipyard personnel. The constraints based on structural considerations were:

1. Web depth < 75 x web thickness, to conform with Horne (1958) to ensure stability against web buckling

2. 20 x web thickness < flange width < 32 x web thickness, to conform with Jenkins (1977) to ensure flange width material is really effective in the section modulus.

After any scantling is increased, the subroutine recalculates the section modulus. If it exceeds the required

section modulus, the subroutine returns the scantling dimensions to the main program. Otherwise, it repeats the above procedure until no dimension can be increased without violating a constraint. At that point an error message is printed out that states scantling limits have been exceeded in subroutine "smact2" and the program sends an indicator to the main program that this condition exists so appropriate action can be taken.

3.5 Modification to Allow Hull Weight Calculation of Double-Bottom Tank Barges

The barge design program requires just a few simple modifications to allow for the calculation of the hull weight of double bottom barges. The mid-section plan of the double-bottom barge including the delineation of the dimensions is shown in Figure 3.2. It should be noted that "l4", h for stanchons, and "h1" differ from Figure 3.1 and that the depth of the double bottom ("dbdep") and the head to the inner bottom longitudinals ("h5") are added.

The program logic for double-bottom barges is identical to that for single-skin barges except for the following items:

1. Instead of calculating the scantlings of the bottom transverse girder which is not part of a double-bottom barge, the program calculates the depth of the double bottom in accordance with the rule presented in Table 3.8.

2. The values of "l4", "h2", and h for stanchions are based on the geometry of Figure 3.2 which differs from the

single-skin case of Figure 3.1.

3. After calculating the minimum scantlings required to satisfy the required section modulus for longitudinal bulkhead transverses, the program calculates values for the double-bottom plate thickness ("dbplt"), double-bottom side and center girder thicknesses ("siddbgirdth" and "cendbgirdth"), and head for double-bottom longitudinals ("h5") in accordance with the rules of ABS (1973) and (1978), presented in summary form in Table 3.8.

4. Following 3. above, the program determines the required section modulus for double-bottom longitudinal stiffeners in accordance with rules presented in Table 3.4. Following this, subroutine "smact1" is called using this required section modulus, "dbplt", and the effective plate width (calculated in a manner similar to the other scantlings) as input parameters. After this, the program proceeds to the routine that calculates hull-girder section moduli.

5. In the hull-girder section moduli calculation, proper account is taken of the longitudinal members of the double bottom, i.e., double bottom plate and longitudinals as well as center and side girders.

6. In the hull weight calculation, proper account is taken for the longitudinal members of the double bottom as well as the double-bottom floors which replace the bottom transverse girders of single-skin barges.

3.6 Barge Design Model Results

3.6.1 Input and Output Formats

To run the barge design model the simple input/output program "bargtest1", shown in Figure 3.6, was written. This program allows the user to specify values of the following input variables: (1)

1. Barge Principal Dimensions: length--"l", breadth--"b", depth--"d", draft--"t", and block coefficient--"cb" (2)

2. Range and Increment for Longitudinal Frame Spacings: "minlonspac", "maxlonspac", and "incrlonspac"

3. Range and Increment for Transverse Web Spacings: "minxvrspac", "maxxvrspac", and "incrxvrspac"

4. Barge Internal Structure: "opt". For single-skin tank barges with a center longitudinal bulkhead, set "opt" to zero. For single-skin barges with two equally spaced longitudinal bulkheads and no stanchions, set "opt" to two. Or, for double-bottom barges with center longitudinal bulkheads, set "opt" to three.

5. Maximum "n" Value: "maxn". See section 3.2.7 for an explanation of the use of the variable "n".

---

(1) For more discussion concerning these variables see Section 3.2.1.

(2) If a depth of zero is inputted, the program "bargtest1" will calculate the minimum freeboard allowed by the rules specified in IMCO (1966) and add this amount to the draft to obtain the depth. See Figure 2.8 for the rules used.

6. Output Format: "cycle". For a detailed summary of the hull scantlings for the optimum barge, set "cycle" to zero. For an overall summary of each case of longitudinal and transverse spacing and "n" value considered, set "cycle to one. Or, for an overall summary of only the optimum system, set "cycle" to two.

Examples of the output format types are shown in Figures 3.7-3.9 for a single-skin, center longitudinal bulkhead tank barge of dimensions 550'x85.94'x35.81' (LxBxT) (1) and block coefficient of 0.8. In these examples, longitudinal spacings of 1.00-3.25 feet in 3 inch increments and transverse spacings of 6-10 feet in one foot increments are considered. Figure 3.7 shows the input and the summary hullweight/section modulus output for the optimum barge ("cycle"=2). Figure 3.8 shows the detailed hull scantlings output for the same barge ("cycle"=0). And, figure 3.9 shows the complete summary hull weight/section modulus output for all the different transverse web spacings, longitudinal frame spacings, and "n" values considered ("cycle"=1).

3.6.2 Barge Design Model Output: Tabular and Graphical

The barge design model was run for barges of the follow-

(1) A barge depth of 44.98' was calculated by the program "bargtest1" to satisfy the minimum freeboard requirements specified in IMCO (1966).

ing size and form: (1)

Length:	250'-700' by 75' increments
Length-Breadth Ratio:	5.6-8.0 by 0.2 increments
Breadth-Draft Ratio:	2.2-3.0 by 0.2 increments
Block Coefficient	
Single-Skin:	0.75, 0.80, 0.85
Double-Bottom:	0.80

Tabular and graphical output of the hull steel weight for single-skin, center longitudinal bulkhead tank barges of the above size and form are presented in Tables 3.9-3.11 and Figures 3.10-3.12. Similary, tabular output of the hull steel weight for double-bottom, center longitudinal bulkhead tank barges is presented in Table 3.12. The hull steel weight found was the minimum for barges with longitudinal spacings ranging from 1.00-3.25 feet in 3 inch increments and with transverse web spacings ranging from 5-16 feet in one foot increments.

3.6.3 Regressions Through the Barge Design Model Outputs

In order to reduce the cost of running the drop-and-swap model, regression equations rather than calls to the subprogram "bargdes" are used to estimate barge hull steel weight and length. These regressions, based on the single-skin and double-bottom tank barge runs described in the

(1) Barge depth was determined using the loadline rules specified in Figure 2.8.

previous section, are presented in Table 3.13. They estimate barge hull steel weight and deadweight as a function of barge length, length-breadth ratio (L/B), and breadth-draft ratio (B/T). They also estimate the barge hull steel weight and length as a function of barge deadweight, L/B, and B/T. These regressions are rather good, with "R-Squared" values of 0.999 or better. A comparison of the hullweight regression (function of length, L/B, and B/T) with the barge design model output data is shown in Figures 3.10-3.12. On the graphs, the regression estimates are shown by asterisks while the actual model output data is shown by the curves.

3.6.4 Some Comments on Barge Design Model Output

After examining the barge design model's output for the runs described in Section 3.6.2, the following general observations can be made:

1. Single-skin barges of less than 350' and all double-bottom barges have bottom hull section moduli significantly greater than the minimum value specified by the ABS rules. This is because the ABS rules pertaining to the section moduli of the individual bottom members result in very heavy bottom scantlings.

2. Generally, the optimum transverse and longitudinal spacings of longer barges are wider than for shorter barges.

3. Barges of greater than 650', especially with large depths, often result in member scantlings exceeding the

constraints specified in the model. Barges of these forms will probably need to have a tank-ship internal structure. That is, they will probably require two longitudinal bulkheads to be used to create side tanks with very deep, built-up webs.

4. The addition of a double-bottom increases the hull steel weight of single-skin tank barges of lengths 750', 625', 550', 475', 400', 325', and 250' approximately 18%, 14%, 8%, 6%, 4%, 1%, and 0%, respectively.

3.6.5 Model Validation

It was originally planned to validate the barge design model by comparing its estimates with the actual hull steel weight values of tank barges currently in operation. However, the detailed design data necessary to accomplish this validation could not be obtained. Although length, breadth, draft, deadweight, and hull steel weight values for several single-skin tank barges are available, the scantling length and block coefficient values are not. (1) Unfortunately, these two data elements are essential for validating and calibrating the model. Nevertheless, a partial validation can be made by selecting values for these variables that yield a cargo deadweight capacity equal to that of the barge. This

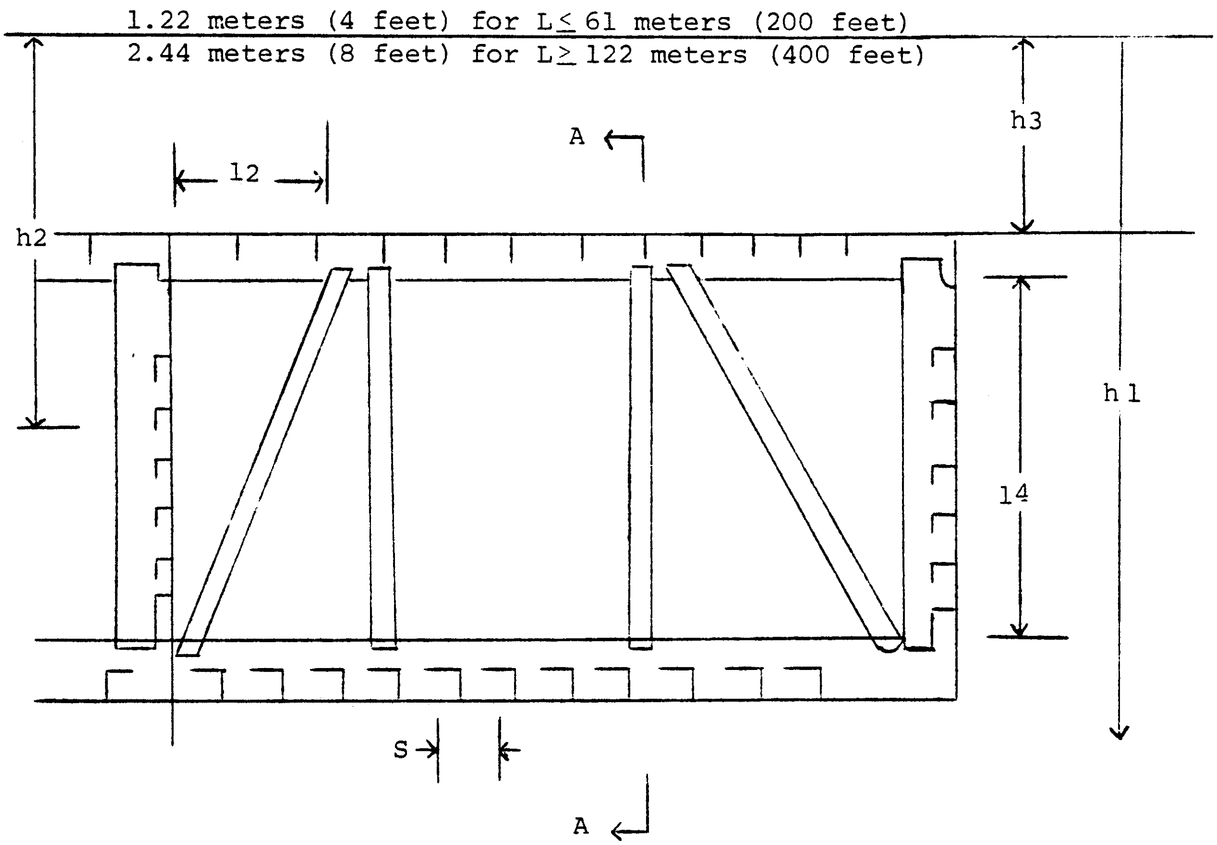
(1) The scantling length is the length used in classification society rules for the calculation of hull girder and member section moduli. This length depends on the type of linkage and depth of notch. See MARAD (1979) for a detailed explanation.

has been done for four tank barges ranging from 4,000 to 40,000 DWT as shown in Table 3.14. There it is seen that when the length between perpendiculars is used for the scantling length and the block coefficient is adjusted to yield the barge deadweight, then the model under/overestimates hull steel weight by 11/22%. And when a block coefficient of 0.90 is used and the hull scantling length is adjusted to yield the barge deadweight, then the model under/overestimates by 21/6%.

These results indicate that the barge design model will produce results of sufficient accuracy for the purpose of cost estimation in the drop-and-swap model. However, before its results can be trusted for preliminary design purposes, it should be calibrated and adjusted by comparing its hull steel weight estimates with the actual hull steel weights of barges for which all particulars are known including notch size, scantling length, block coefficient, and percentage lightweight, high strength steel used in its construction. (1)

---

(1) The barge design model assumes that the barge is totally fabricated out of mild steel. Therefore, if high strength, light weight steel is used, then the model's hull steel weight estimate must be reduced in proportion to the amount of mild steel replaced by the light weight steel.



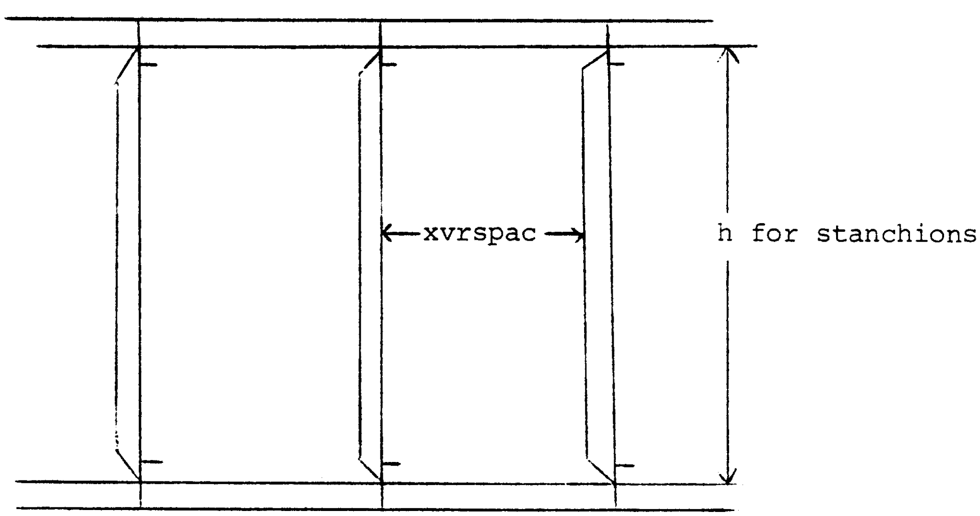


FIGURE 3.1

SCANTLING ARRANGEMENT FOR SINGLE-SKIN TANK BARGE

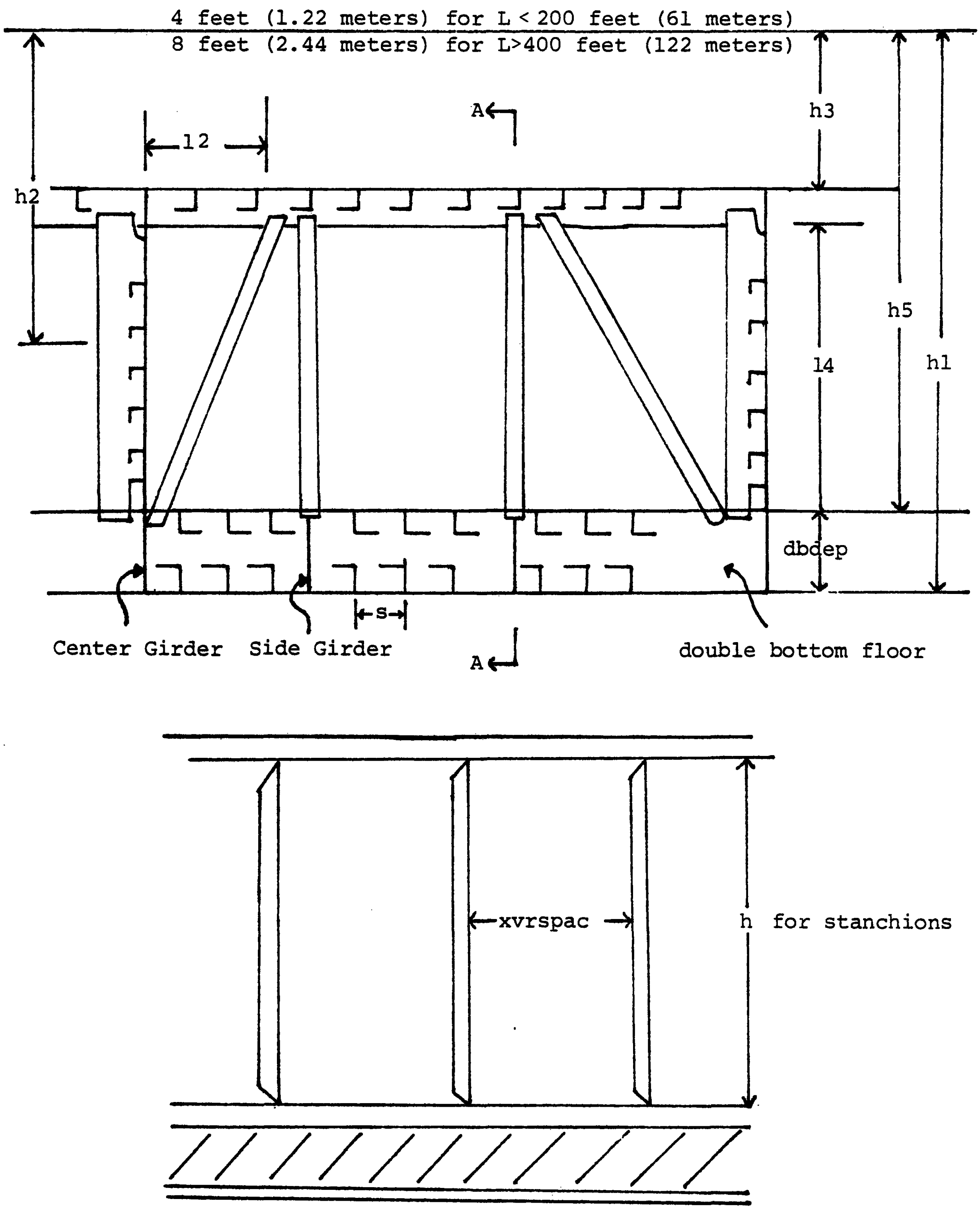


FIGURE 3.2

SCANTLING ARRANGEMENT FOR DOUBLE-BOTTOM TANK BARGE

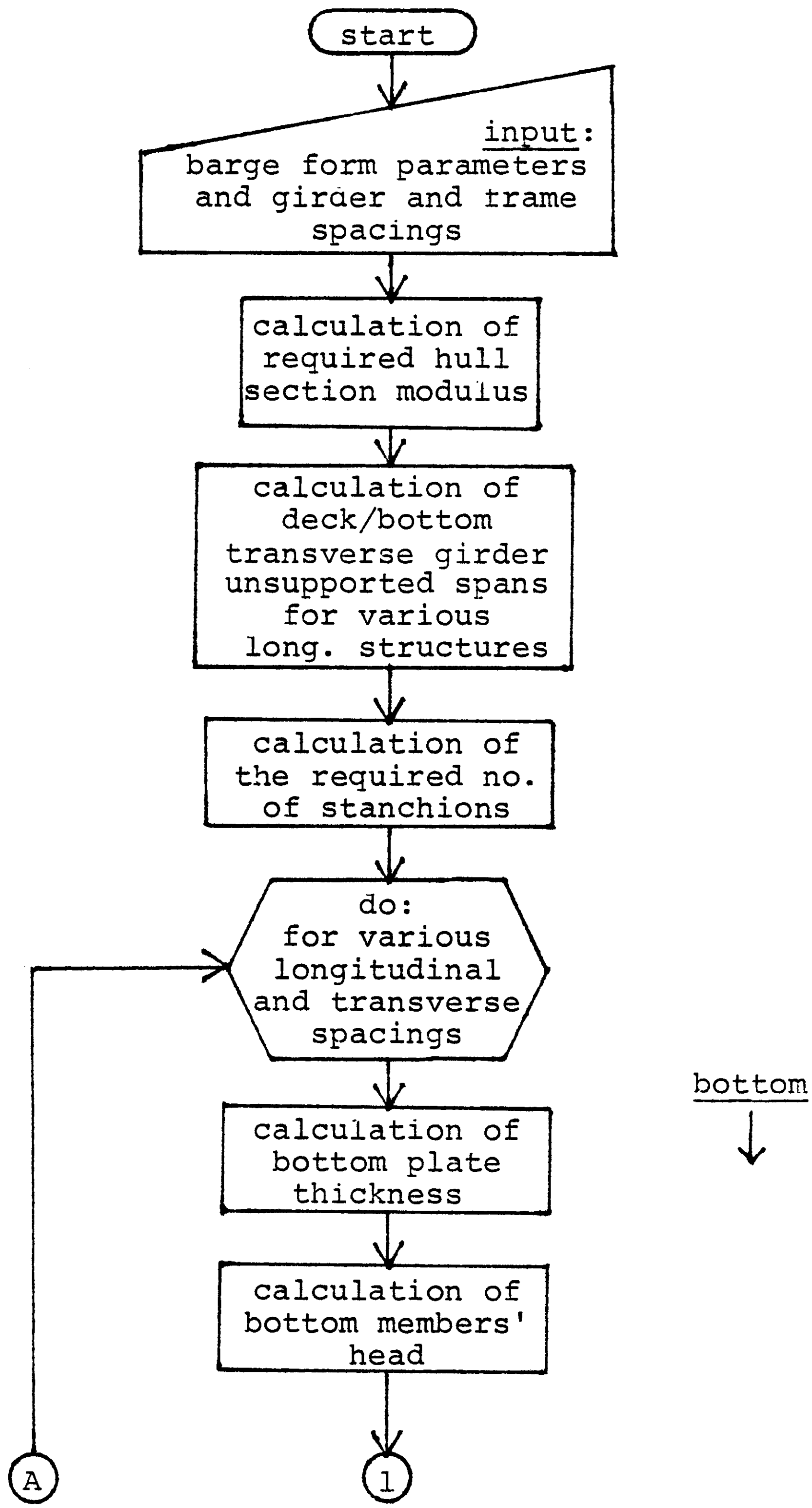


FIGURE 3.3

SUMMARY FLOWCHART OF BARGE DESIGN SUBPROGRAM

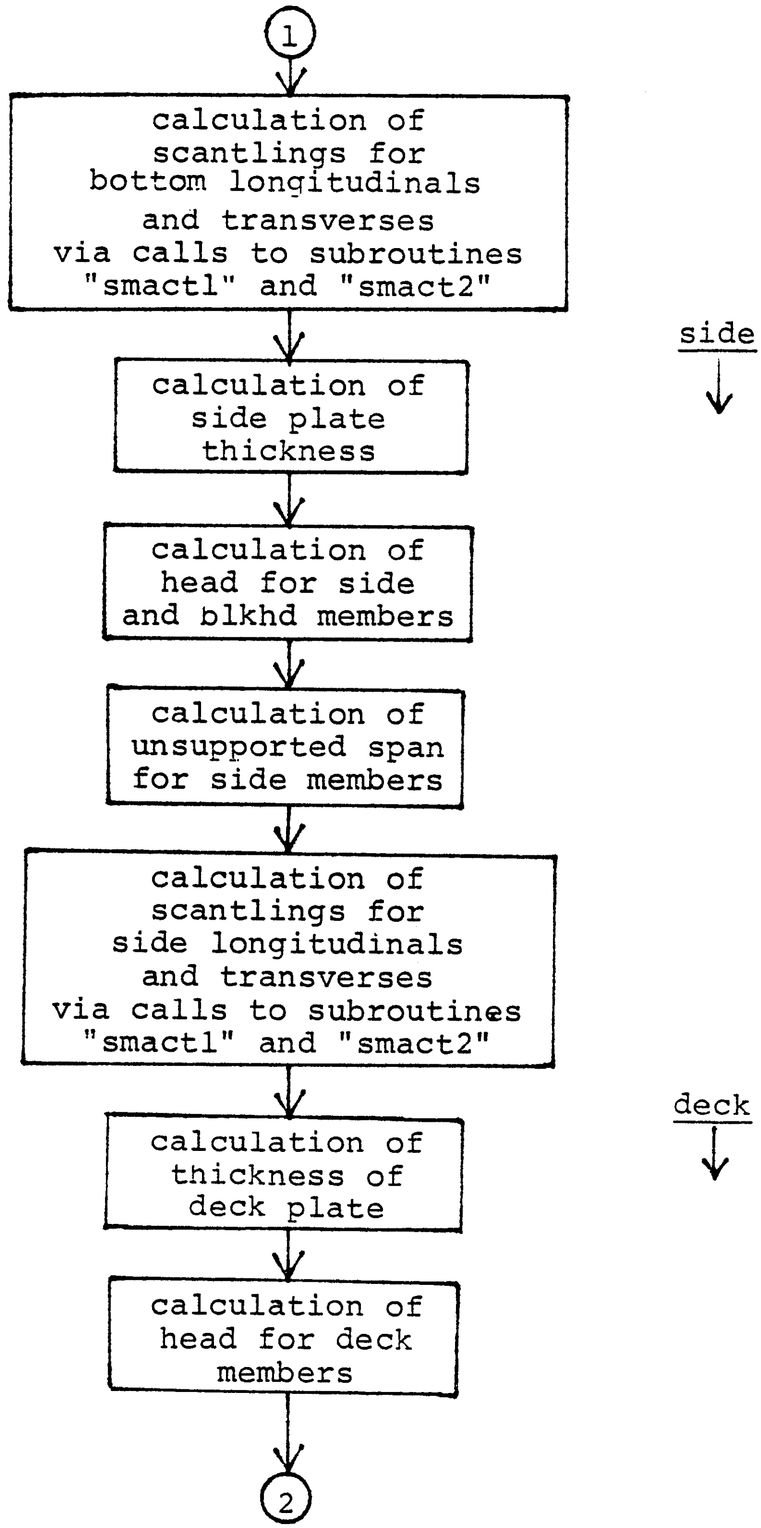


FIGURE 3.3...(Continued)

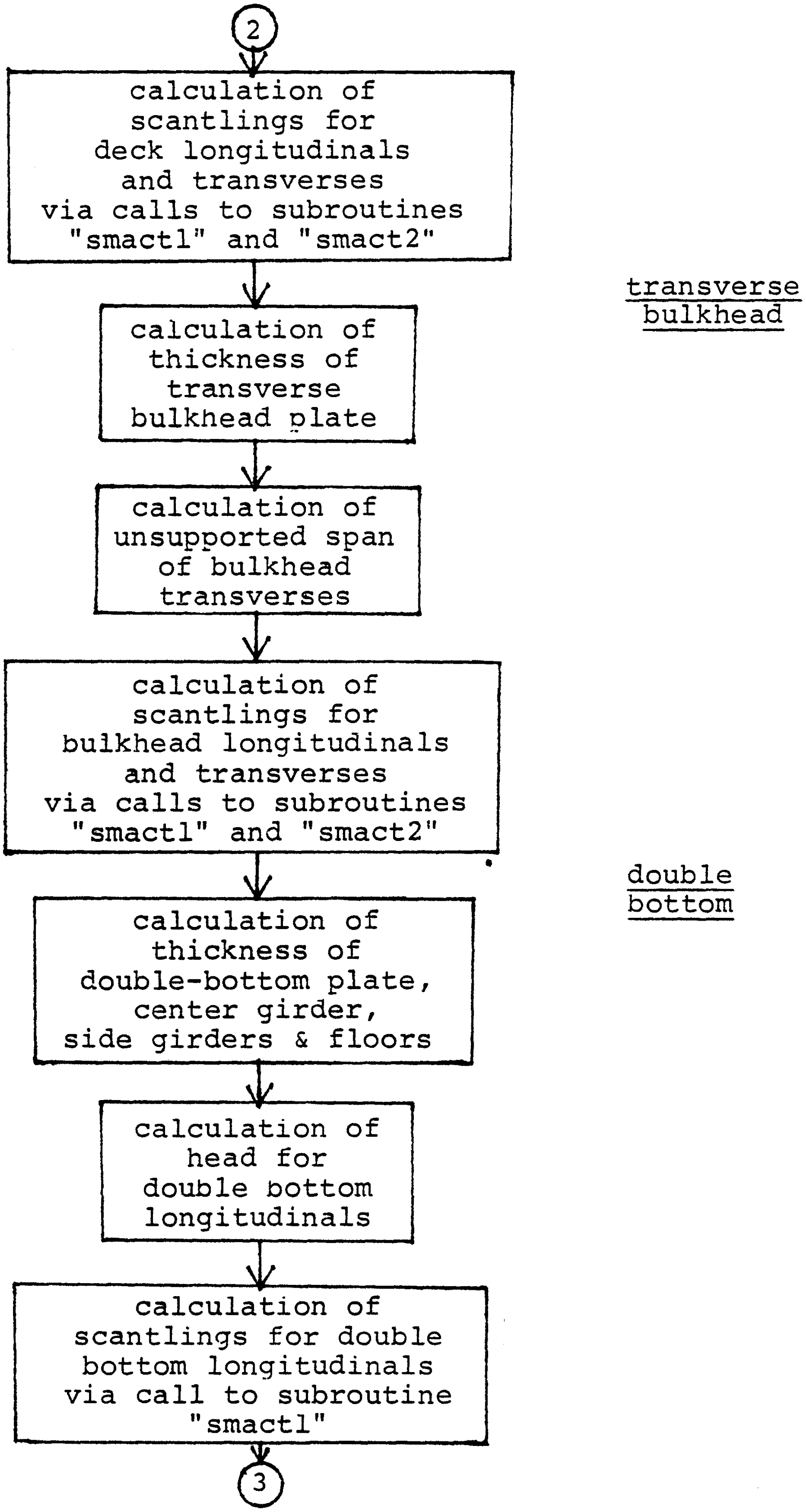


FIGURE 3.3...(Continued)

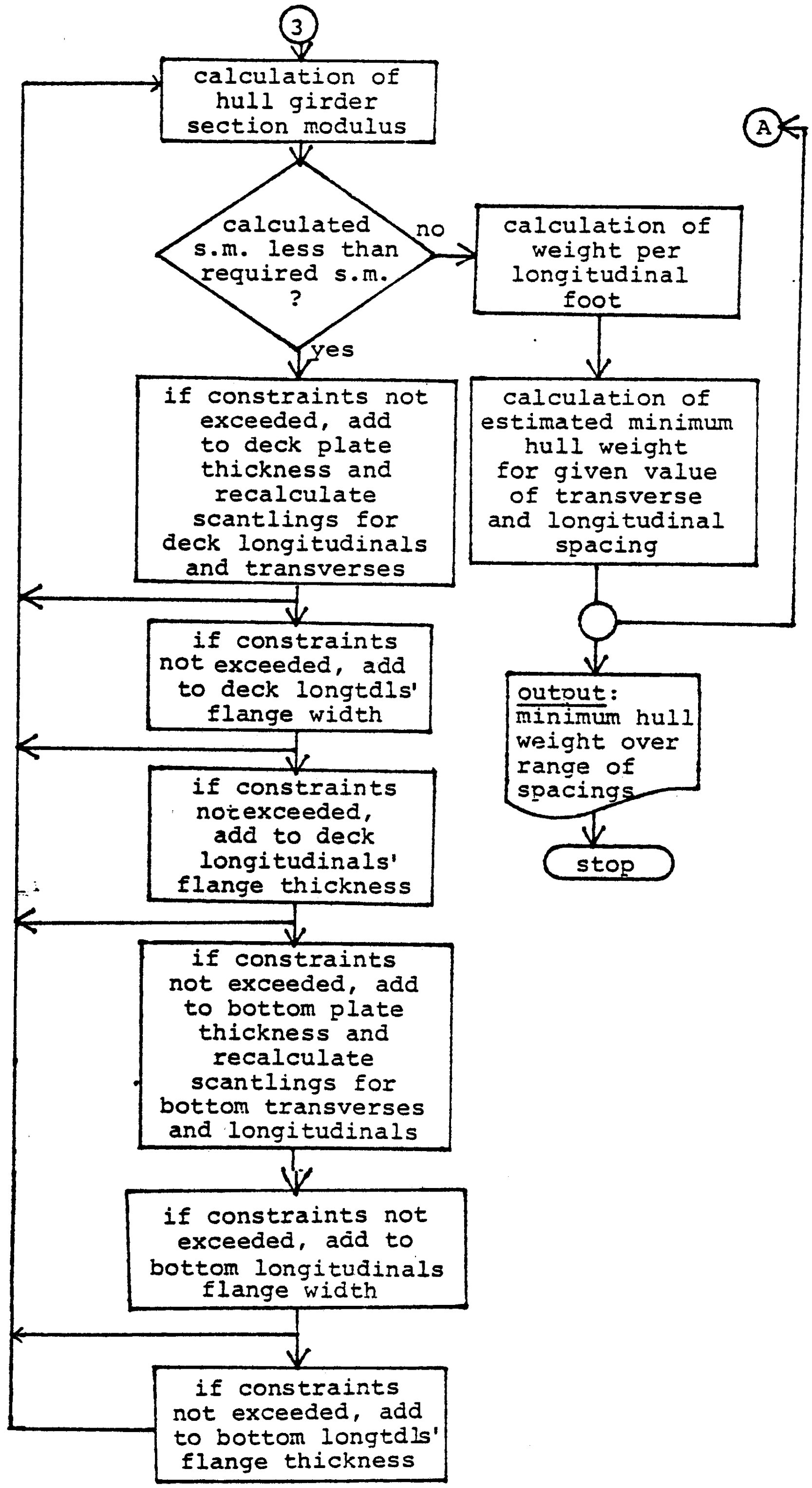


FIGURE 3.3...(Continued)

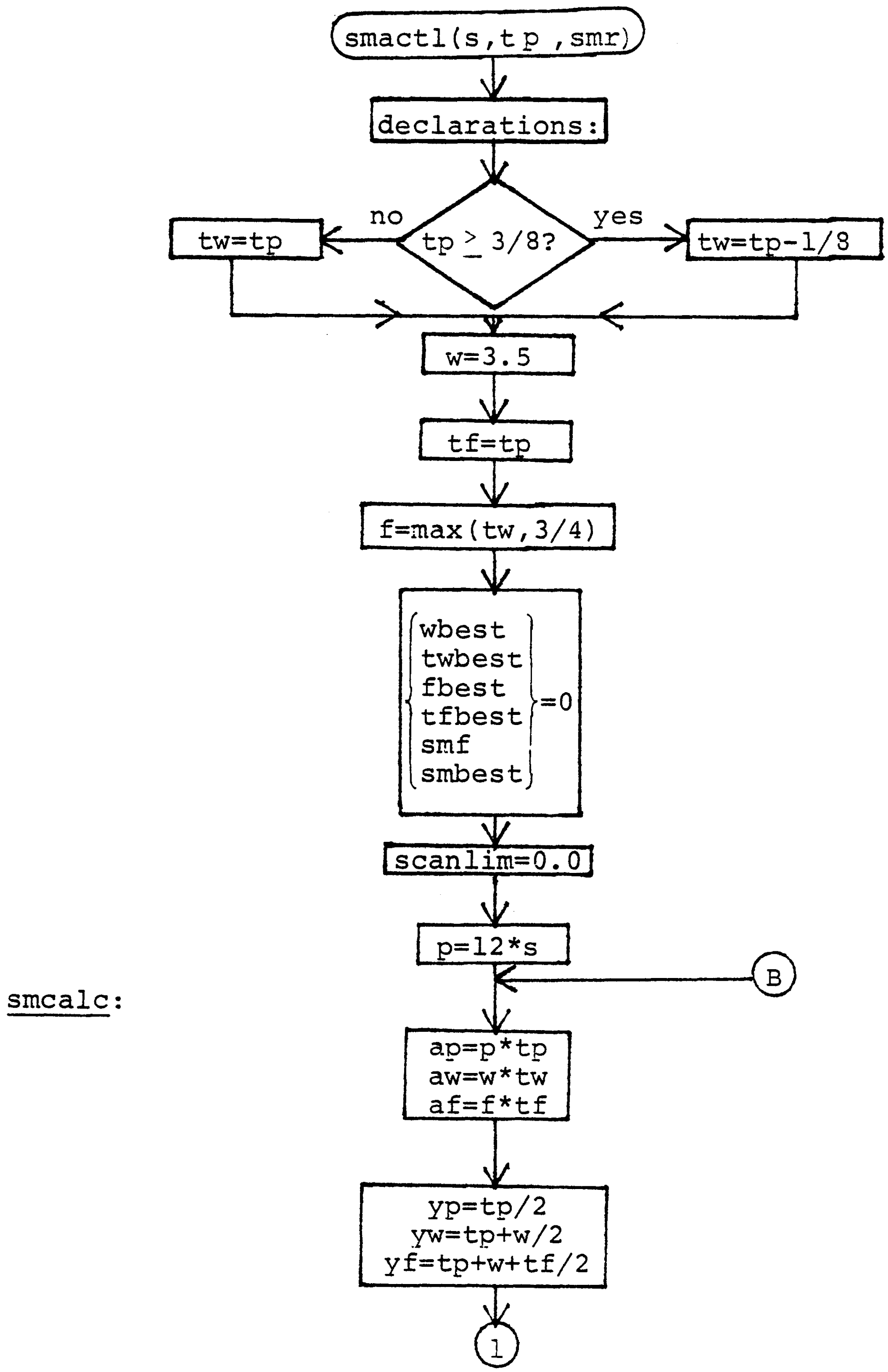


FIGURE 3.4

FLOWCHART OF SUBROUTINE "smactl"

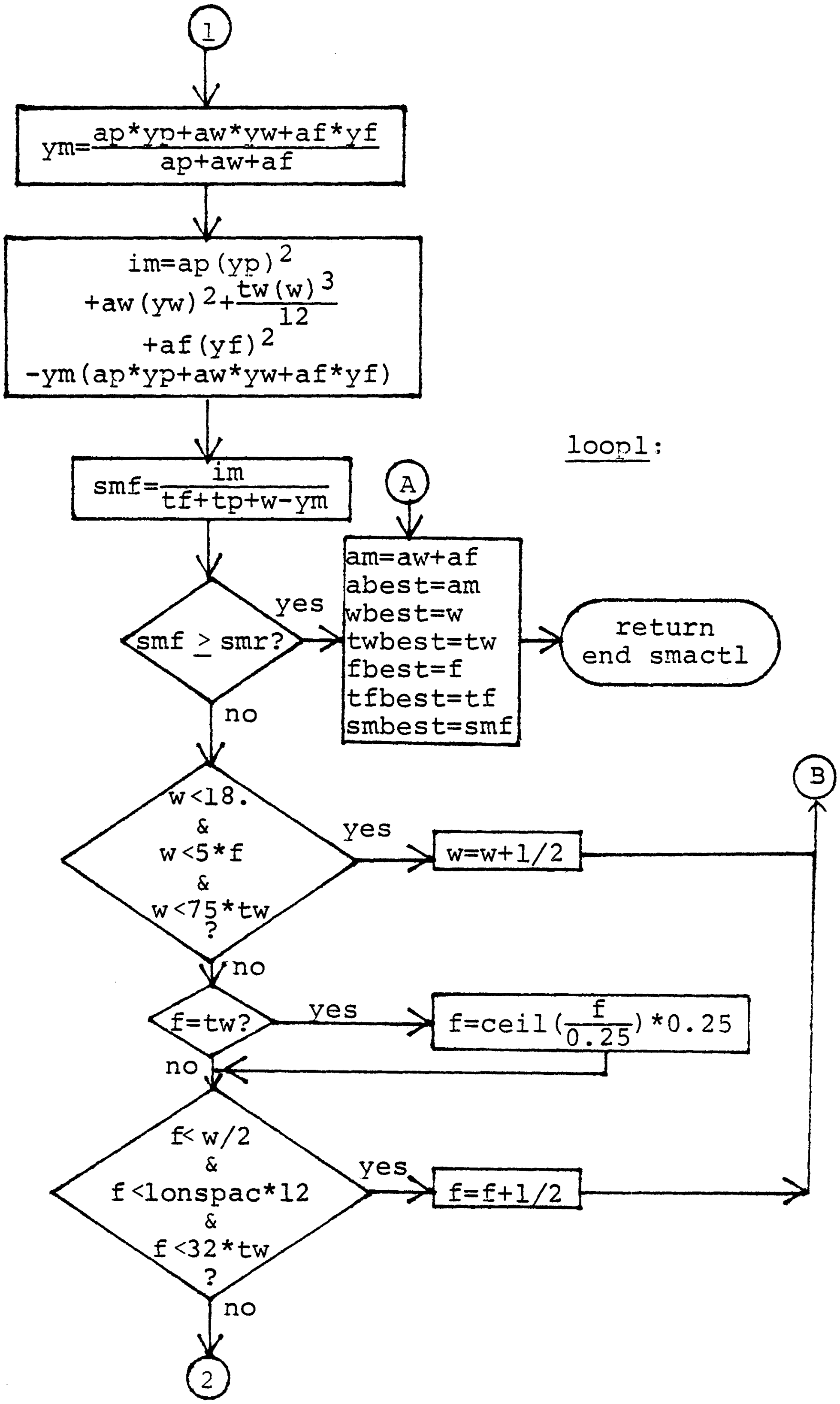


FIGURE 3.4...(Continued)

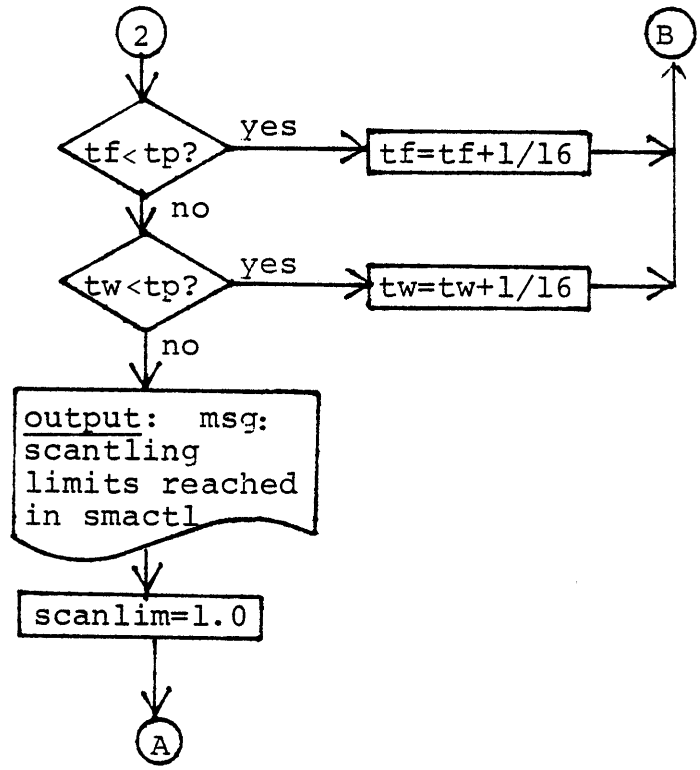


FIGURE 3.4...(Continued)

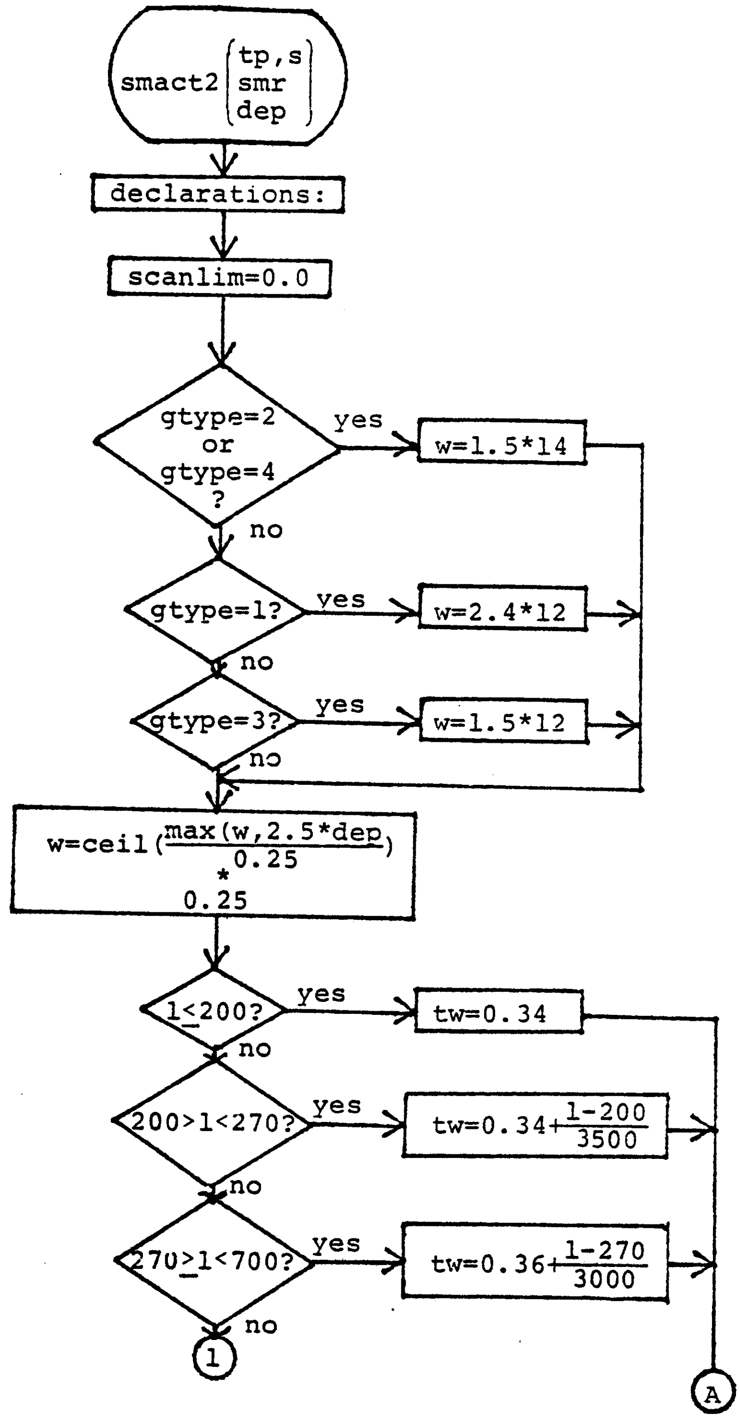


FIGURE 3.5
FLOWCHART OF SUBROUTINE "smact2"

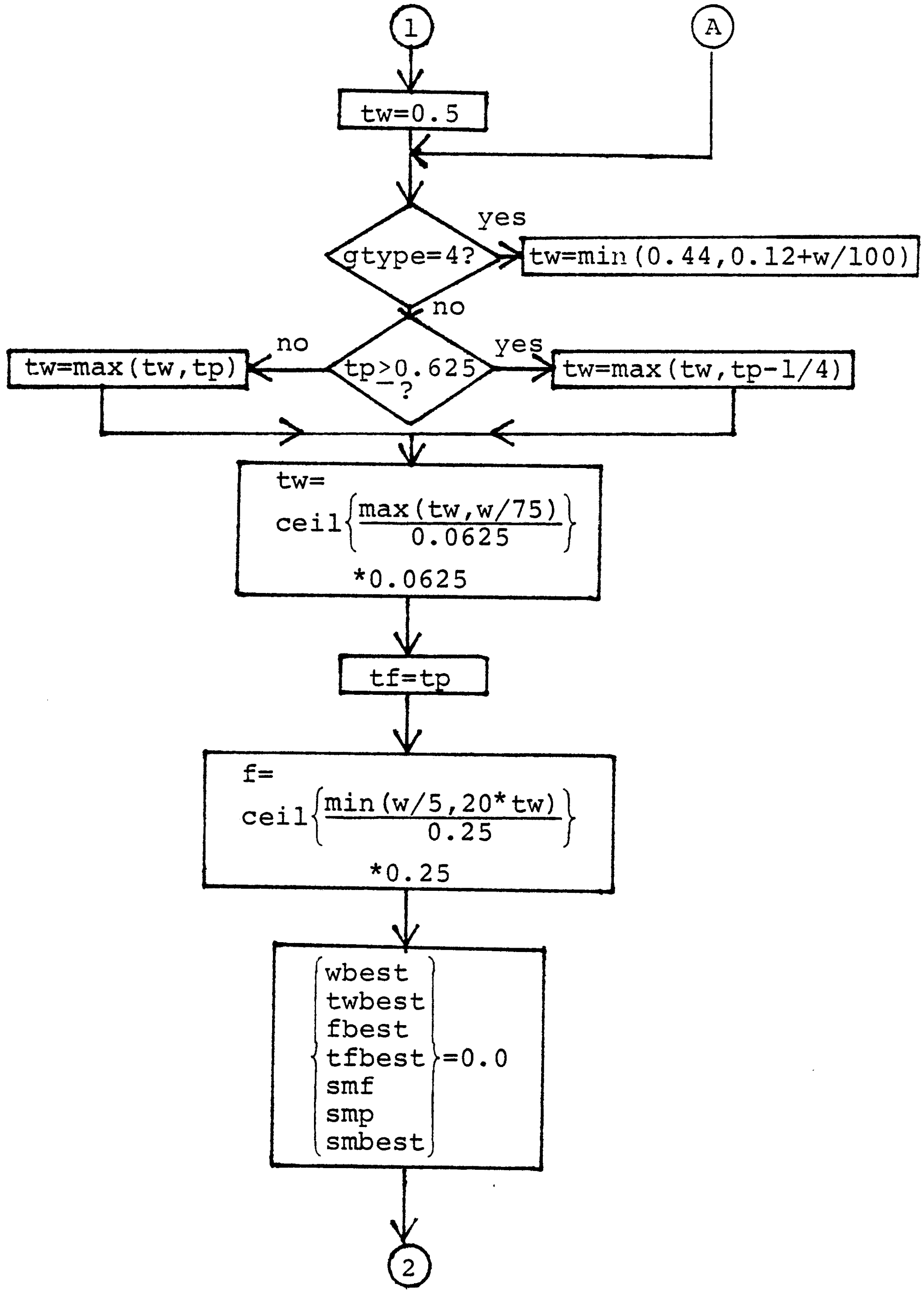


FIGURE 3.5...(Continued)

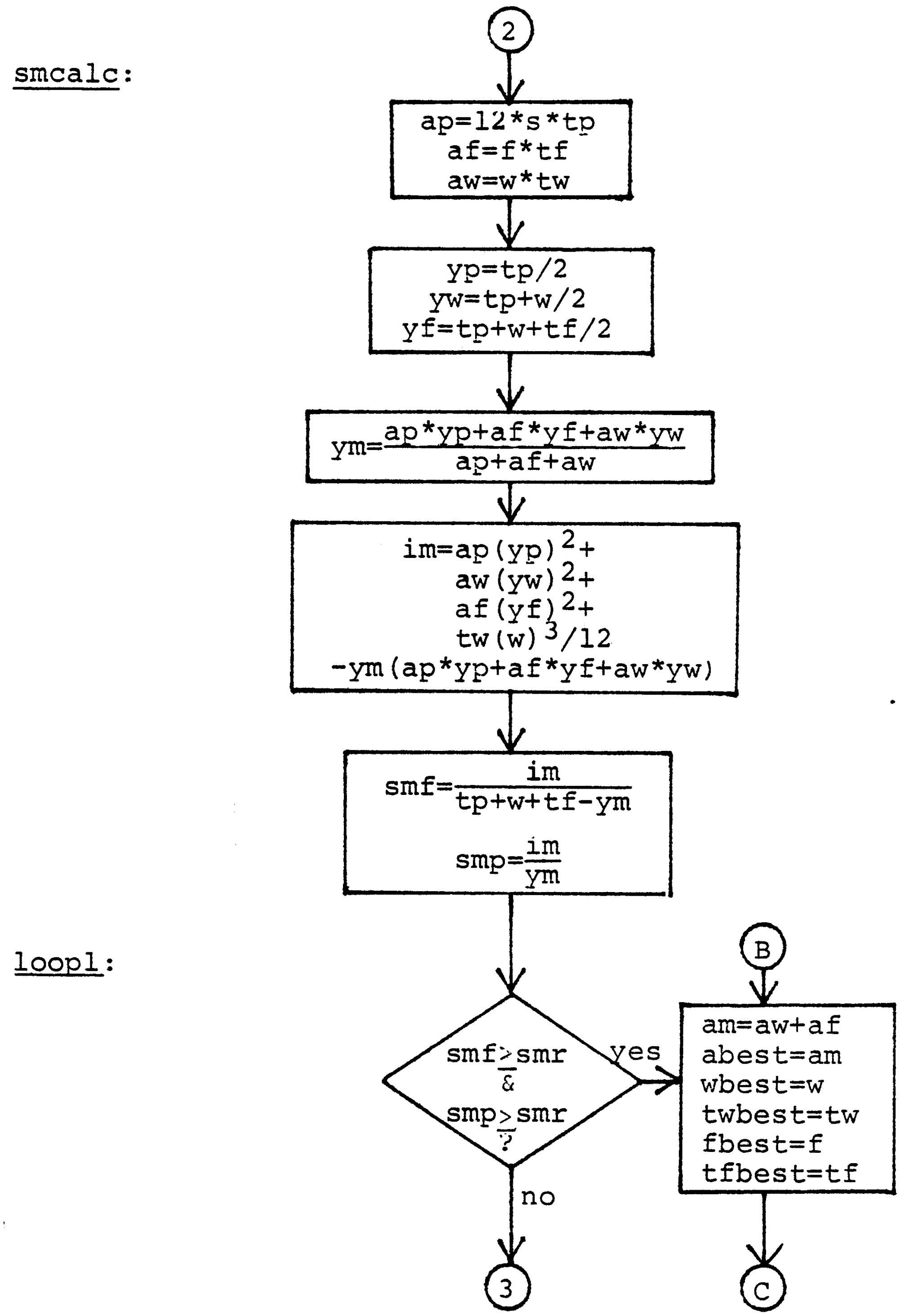


FIGURE 3.5...(Continued)

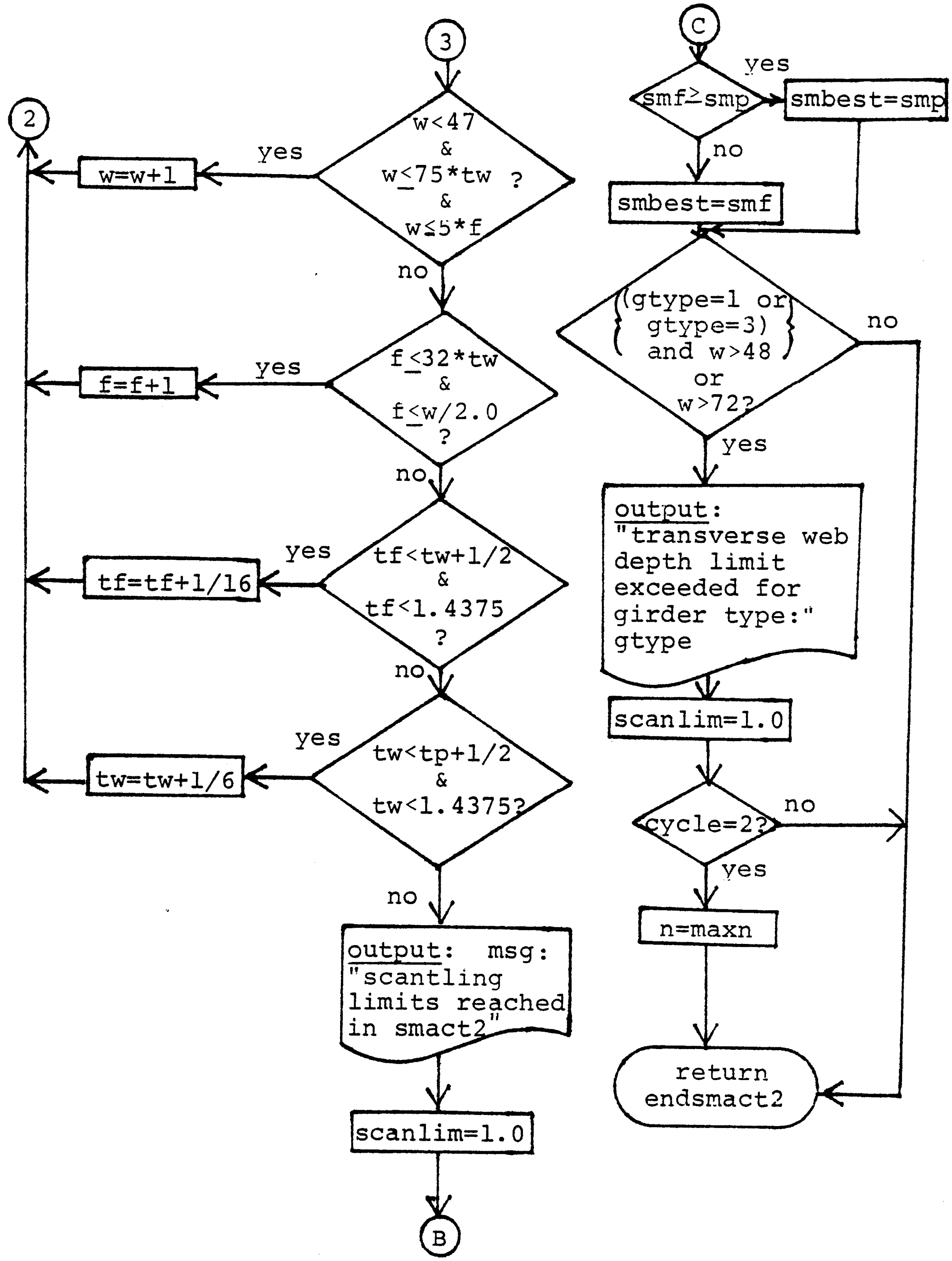


FIGURE 3.5...(Continued)

```
bargtest1: proc;
          dcl(l,b,d,t,cb,minlonspac,maxlonspac,incrlonspac,minxvrspac,maxxvrspac,incrxvrspac,coeff(66),llcoef(66)) dec float;
          dcl (minfbd,fbd,ldfact,shrfact,mind,displ,vol,volwt,outfitwt,dwt,minwt,bestlonspac,bestxvrspac) dec float;
          dcl (cycle,countr,opt,minn,maxn,bestn) bin fixed;
          dcl (sysin,absfact) file stream input;
          dcl sysprint file stream output;
dcl bargdes6 entry(dec float,dec float,dec float,dec float,dec float,dec float,dec float,dec float,
         dec float,dec float,dec float,bin fixed,bin fixed,dim(66) dec float,bin fixed,bin fixed,dec float,
         dec float,dec float,bin fixed) external;
          open file(absfact) title("vfile_ abscoeff.data");
          get file(absfact) list(coeff);
          get  file(absfact) list(llcoef);
          close file(absfact);
 return:
          put skip(3) list("Please specify values for the following variables:");
          put skip list("minlonspac, maxlonspac, incrlonspac: ");
          put skip;
          get  list(minlonspac,maxlonspac,incrlonspac);
          put skip list("minxvrspac, maxxvrspac, incrxvrspac: ");
          put skip;
          get  list(minxvrspac,maxxvrspac,incrxvrspac);
          put skip list("l,b,d,t,cb: ");
          put skip;
          get  list(l,b,d,t,cb);
          put skip list("maxn, cycle, opt:");
          put skip;
          get list(maxn,cycle,opt);
          minn = 0;
put skip(5) list("LENGTH BEAM DEPTH DRAFT L/B B/T L/D  T/D HULL-WT OUTFIT-WT LT-WT DISPL  DWT  FRBD  VOLWT");
     if l=750. then minfbd=llcoef(66);
               else do;
                    countr=ceil((l-99.99999)/10.);
                    minfbd=llcoef(countr) + (llcoef(countr+1)-llcoef(countr))*(l-90.-10.*countr)/10.;
               end;
     minfbd = minfbd*(cb + 0.68)/1.38;
     if (l/(minfbd/12. + t)) < 15. then do;
          if l<393.6 then ldfact = (t + minfbd/12. - l/15.)*(l/131.2)/(1-l/(12.*131.2));
                    else ldfact = (t + minfbd/12. - l/15.)*4.;
     end;
     else ldfact = 0.;
     shrfact = 0.0375*l + 3.75;
     fbd = 0.75*(minfbd + ldfact + shrfact);
     mind=(t + fbd/12.);
     if d=0. then d=mind;
     displ = l*b*t*cb/35.;
call bargdes6(l,b,d,t,cb,minlonspac,maxlonspac,incrlonspac,minxvrspac,maxxvrspac,incrxvrspac,minn,maxn,
               coeff,cycle,opt,minwt,bestlonspac,bestxvrspac,bestn);
     vol = l*b*t*cb + l*b*(d-t) - (minwt/490.);
     volwt = vol/42.;
     outfitwt = ((400. + (l-290.)*1.7) -230.)*0.88;
      dwt = displ -minwt - outfitwt;
put skip edit(l,b,d,t,l/b,b/t,l/d,t/d,minwt,outfitwt,minwt+outfitwt,displ,dwt,fbd,volwt)
               (col(1),f(3),f(8,2),2 f(5,1),f(4,1),f(5,2),f(5,1),f(5,2),f(6),x(5),f(4),x(1),f(7),2 f(6),f(6,2),f(6));
if cycle = 1 | cycle=2 then do;
     put page;
     go to return;
end;
end bargtest1;
```

FIGURE 3.6

LISTING OF BARGE DESIGN MODEL INPUT/OUTPUT PROGRAM "bargtest1"

```
bargtest1


Please specify values for the following variables:
minlonspac, maxlonspac, incrlonspac:
1.00,3.25,0.25,

minxvrspac, maxxvrspac, incrxvrspac:
5.,12.,1.,

l,b,d,t,cb:
550.0,85.9375,0.,35.8073,0.80,

maxn, cycle, opt:
7,2,0,



LENGTH BEAM  DEPTH DRAFT L/B B/T L/D  T/D HULL-WT OUTFIT-WT LT-WT DISPL  DWT  FRBD  VOLWT
550    85.94 45.0 35.8 6.4 2.40 12.2 0.80  5016      539    5555 38685 33130110.08 42560
\014
```

FIGURE 3.7

SAMPLE INPUT/OUTPUT OF THE BARGE DESIGN MODEL: SUMMARY OUTPUT FOR THE OPTIMUM BARGE

```
bargtest1


Please specify values for the following variables:
minlonspac, maxlonspac, incrlonspac:
2.25,2.25,1.,

minxvrspac, maxxvrspac, incrxvrspac:
9.,9.,1.,

l,b,d,t,cb:
550.0,85.9375,0.,35.8073,0.80,

maxn, cycle, opt:
2,0,0,



LENGTH BEAM DEPTH DRAFT L/B B/T L/D  T/D HULL-WT OUTFIT-WT LT-WT DISPL  DWT  FRBD  VOLWT
 length=550  beam= 85.94  depth=44.98   draft=35.81  required modulus= 51167

 longitudinal frame spacing =  2.25  transverse web spacing =  9.00
 bottom modulus= 54031  deck modulus= 51549  estimated bare hull weight=  5016

                    summary of principal scantlings

 bottom shell - 0.6875 in. side shell - 0.6250 in.  bulkhead plating - 0.5625 in.  deck plate - 0.8125
 bottom longitudinals: flange-    2.75 x 0.6875;   web- 13.50 x 0.5625
 bottom transverses: flange-      7.25 x 0.6875;   web- 35.50 x 0.5000
 side longitudinals: flange-      2.25 x 0.6250;   web- 10.00 x 0.5000
 side transverses: flange-       26.75 x 0.8750;   web- 58.50 x 0.8125
 bulkhead longitudinals: flange-  2.25 x 0.5625;   web-  9.50 x 0.4375
 bulkhead transverses: flange-   26.25 x 0.8750;   web- 60.50 x 0.8125
 deck longitudinals: flange-      1.25 x 0.8125;   web-  4.00 x 0.6875
 deck transverses: flange-        4.00 x 0.8125;   web- 19.25 x 0.5625bestlonspac= 2.250000000e+000 bestxvrspac= 9.000000000e+000
bestn=         2    smreq= 5.116718750e+004        bestmodbot= 5.403096746e+004  bestmoddk= 5.154864626e+004;
550   85.94 45.0 35.8 6.4 2.40 12.2 0.80  5016      539    5555 38685 33130110.08 42560
```

FIGURE 3.8

EXAMPLE OF DETAILED HULL SCANTLING OUTPUT FROM BARGE DESIGN SUBMODEL

```
bargtest1

Please specify values for the following variables:
minlonspac, maxlonspac, incrlonspac:
1.50,2.75,0.25,

minxvrspac, maxxvrspac, incrxvrspac:
7.,10.,1.,

l,b,d,t,cb:
550.0,85.9375,0.,35.8073,0.80,

maxn, cycle, opt:
5,1,0,


LENGTH BEAM DEPTH DRAFT L/B B/T L/D  T/D HULL-WT OUTFIT-WT LT-WT DISPL  DWT  FRBD  VOLWT
 length=550  beam= 85.94  depth=44.98   draft=35.81  required modulus= 51167

  longitudinal frame spacing =  1.50  transverse web spacing =  7.00
  bottom modulus= 58806  deck modulus= 52220  estimated bare hull weight=  5324

  longitudinal frame spacing =  1.50  transverse web spacing =  7.00
  bottom modulus= 55734  deck modulus= 51815  estimated bare hull weight=  5233

  longitudinal frame spacing =  1.50  transverse web spacing =  7.00
  bottom modulus= 52528  deck modulus= 51352  estimated bare hull weight=  5163

  longitudinal frame spacing =  1.50  transverse web spacing =  7.00
  bottom modulus= 52528  deck modulus= 51352  estimated bare hull weight=  5163

  longitudinal frame spacing =  1.50  transverse web spacing =  8.00
  bottom modulus= 60873  deck modulus= 52506  estimated bare hull weight=  5298

  longitudinal frame spacing =  1.50  transverse web spacing =  8.00
  bottom modulus= 57763  deck modulus= 52100  estimated bare hull weight=  5220

  longitudinal frame spacing =  1.50  transverse web spacing =  8.00
  bottom modulus= 54383  deck modulus= 51617  estimated bare hull weight=  5145

  longitudinal frame spacing =  1.50  transverse web spacing =  8.00
  bottom modulus= 54383  deck modulus= 51617  estimated bare hull weight=  5145

  longitudinal frame spacing =  1.50  transverse web spacing =  9.00
  bottom modulus= 59456  deck modulus= 51568  estimated bare hull weight=  5303

  longitudinal frame spacing =  1.50  transverse web spacing =  9.00
  bottom modulus= 56053  deck modulus= 51885  estimated bare hull weight=  5245

  longitudinal frame spacing =  1.50  transverse web spacing =  9.00
  bottom modulus= 53001  deck modulus= 51402  estimated bare hull weight=  5177

  longitudinal frame spacing =  1.50  transverse web spacing =  9.00
  bottom modulus= 53017  deck modulus= 51379  estimated bare hull weight=  5187

  longitudinal frame spacing =  1.50  transverse web spacing =  9.00
  bottom modulus= 53017  deck modulus= 51379  estimated bare hull weight=  5187

  longitudinal frame spacing =  1.50  transverse web spacing = 10.00
  bottom modulus= 61398  deck modulus= 51839  estimated bare hull weight=  5343
```

FIGURE 3.9

EXAMPLE OF COMPLETE SUMMARY OUTPUT FROM BARGE DESIGN SUBPROGRAM

longitudinal frame spacing = 1.50 transverse web spacing = 10.00
bottom modulus= 57680 deck modulus= 51333 estimated bare hull weight= 5260

longitudinal frame spacing = 1.50 transverse web spacing = 10.00
bottom modulus= 54469 deck modulus= 51617 estimated bare hull weight= 5208

longitudinal frame spacing = 1.50 transverse web spacing = 10.00
bottom modulus= 54469 deck modulus= 51617 estimated bare hull weight= 5208

longitudinal frame spacing = 1.75 transverse web spacing = 7.00
bottom modulus= 58103 deck modulus= 51408 estimated bare hull weight= 5219

longitudinal frame spacing = 1.75 transverse web spacing = 7.00
bottom modulus= 55130 deck modulus= 51758 estimated bare hull weight= 5144

longitudinal frame spacing = 1.75 transverse web spacing = 7.00
bottom modulus= 52032 deck modulus= 51302 estimated bare hull weight= 5077

longitudinal frame spacing = 1.75 transverse web spacing = 7.00
bottom modulus= 52382 deck modulus= 53535 estimated bare hull weight= 5140

longitudinal frame spacing = 1.75 transverse web spacing = 7.00
bottom modulus= 52382 deck modulus= 53535 estimated bare hull weight= 5140

longitudinal frame spacing = 1.75 transverse web spacing = 8.00
bottom modulus= 59377 deck modulus= 51603 estimated bare hull weight= 5255

longitudinal frame spacing = 1.75 transverse web spacing = 8.00
bottom modulus= 56188 deck modulus= 51177 estimated bare hull weight= 5174

longitudinal frame spacing = 1.75 transverse web spacing = 8.00
bottom modulus= 53499 deck modulus= 51508 estimated bare hull weight= 5128

longitudinal frame spacing = 1.75 transverse web spacing = 8.00
bottom modulus= 53873 deck modulus= 53734 estimated bare hull weight= 5189

longitudinal frame spacing = 1.75 transverse web spacing = 8.00
bottom modulus= 53873 deck modulus= 53734 estimated bare hull weight= 5189

longitudinal frame spacing = 1.75 transverse web spacing = 9.00
bottom modulus= 61597 deck modulus= 51583 estimated bare hull weight= 5380

longitudinal frame spacing = 1.75 transverse web spacing = 9.00
bottom modulus= 58417 deck modulus= 52106 estimated bare hull weight= 5323

longitudinal frame spacing = 1.75 transverse web spacing = 9.00
bottom modulus= 55028 deck modulus= 51615 estimated bare hull weight= 5251

longitudinal frame spacing = 1.75 transverse web spacing = 9.00
bottom modulus= 51958 deck modulus= 51834 estimated bare hull weight= 5195

longitudinal frame spacing = 1.75 transverse web spacing = 9.00
bottom modulus= 52339 deck modulus= 54096 estimated bare hull weight= 5256

longitudinal frame spacing = 1.75 transverse web spacing = 9.00
bottom modulus= 52339 deck modulus= 54096 estimated bare hull weight= 5256

longitudinal frame spacing = 1.75 transverse web spacing = 10.00
bottom modulus= 59872 deck modulus= 51373 estimated bare hull weight= 5311

longitudinal frame spacing = 1.75 transverse web spacing = 10.00
bottom modulus= 59872 deck modulus= 51373 estimated bare hull weight= 5311

longitudfnal frame spacing = 2.00 transverse web spacing = 7.00
bottom modulus= 57551 deck modulus= 51363 estimated bare hull weight= 5248

longitudinal frame spacing = 2.00 transverse web spacing = 7.00
bottom modulus= 54727 deck modulus= 51601 estimated bare hull weight= 5174

longitudinal frame spacing = 2.00 transverse web spacing = 7.00
bottom modulus= 51687 deck modulus= 51755 estimated bare hull weight= 5120

longitudinal frame spacing = 2.00 transverse web spacing = 7.00
bottom modulus= 51954 deck modulus= 53344 estimated bare hull weight= 5169

longitudinal frame spacing = 2.00 transverse web spacing = 7.00
bottom modulus= 51954 deck modulus= 53344 estimated bare hull weight= 5169

longitudinal frame spacing = 2.00 transverse web spacing = 8.00
bottom modulus= 59274 deck modulus= 52163 estimated bare hull weight= 5204

longitudinal frame spacing = 2.00 transverse web spacing = 8.00
bottom modulus= 56102 deck modulus= 51707 estimated bare hull weight= 5123

longitudinal frame spacing = 2.00 transverse web spacing = 8.00
bottom modulus= 53253 deck modulus= 51250 estimated bare hull weight= 5059

longitudinal frame spacing = 2.00 transverse web spacing = 8.00
bottom modulus= 53253 deck modulus= 51250 estimated bare hull weight= 5059

longitudinal frame spacing = 2.00 transverse web spacing = 9.00
bottom modulus= 60499 deck modulus= 51238 estimated bare hull weight= 5202

longitudinal frame spacing = 2.00 transverse web spacing = 9.00
bottom modulus= 57691 deck modulus= 52587 estimated bare hull weight= 5167

longitudinal frame spacing = 2.00 transverse web spacing = 9.00
bottom modulus= 54706 deck modulus= 52106 estimated bare hull weight= 5103

longitudinal frame spacing = 2.00 transverse web spacing = 9.00
bottom modulus= 51530 deck modulus= 51547 estimated bare hull weight= 5035

longitudinal frame spacing = 2.00 transverse web spacing = 9.00
bottom modulus= 51530 deck modulus= 51547 estimated bare hull weight= 5035

longitudinal frame spacing = 2.00 transverse web spacing = 10.00
bottom modulus= 59187 deck modulus= 51552 estimated bare hull weight= 5302

longitudinal frame spacing = 2.00 transverse web spacing = 10.00
bottom modulus= 59187 deck modulus= 51552 estimated bare hull weight= 5302

longitudinal frame spacing = 2.25 transverse web spacing = 7.00
bottom modulus= 57013 deck modulus= 51561 estimated bare hull weight= 5289

longitudinal frame spacing = 2.25 transverse web spacing = 7.00
bottom modulus= 54100 deck modulus= 51694 estimated bare hull weight= 5212

longitudinal frame spacing = 2.25 transverse web spacing = 7.00
bottom modulus= 54100 deck modulus= 51694 estimated bare hull weight= 5212

longitudinal frame spacing = 2.25 transverse web spacing = 8.00
bottom modulus= 58339 deck modulus= 51673 estimated bare hull weight= 5180

longitudinal frame spacing = 2.25 transverse web spacing = 8.00
bottom modulus= 55628 deck modulus= 51263 estimated bare hull weight= 5109

longitudinal frame spacing = 2.25 transverse web spacing = 8.00
bottom modulus= 52567 deck modulus= 51330 estimated bare hull weight= 5051

longitudinal frame spacing = 2.25 transverse web spacing = 8.00
bottom modulus= 52976 deck modulus= 53630 estimated bare hull weight= 5114

FIGURE 3.9...(Continued)

longitudinal frame spacing = 2.25 transverse web spacing = 8.00
bottom modulus= 52976 deck modulus= 53630 estimated bare hull weight= 5114

longitudinal frame spacing = 2.25 transverse web spacing = 9.00
bottom modulus= 60320 deck modulus= 52519 estimated bare hull weight= 5169

longitudinal frame spacing = 2.25 transverse web spacing = 9.00
bottom modulus= 57036 deck modulus= 52039 estimated bare hull weight= 5086

longitudinal frame spacing = 2.25 transverse web spacing = 9.00
bottom modulus= 54031 deck modulus= 51549 estimated bare hull weight= 5016

longitudinal frame spacing = 2.25 transverse web spacing = 9.00
bottom modulus= 54031 deck modulus= 51549 estimated bare hull weight= 5016

longitudinal frame spacing = 2.25 transverse web spacing = 10.00
bottom modulus= 58471 deck modulus= 51298 estimated bare hull weight= 5250

longitudinal frame spacing = 2.25 transverse web spacing = 10.00
bottom modulus= 55071 deck modulus= 51276 estimated bare hull weight= 5200

longitudinal frame spacing = 2.25 transverse web spacing = 10.00
bottom modulus= 55071 deck modulus= 51276 estimated bare hull weight= 5200

longitudinal frame spacing = 2.50 transverse web spacing = 7.00
bottom modulus= 56693 deck modulus= 51494 estimated bare hull weight= 5299

longitudinal frame spacing = 2.50 transverse web spacing = 7.00
bottom modulus= 53806 deck modulus= 51551 estimated bare hull weight= 5220

longitudinal frame spacing = 2.50 transverse web spacing = 7.00
bottom modulus= 53806 deck modulus= 51551 estimated bare hull weight= 5220

longitudinal frame spacing = 2.50 transverse web spacing = 8.00
bottom modulus= 58393 deck modulus= 52170 estimated bare hull weight= 5219

longitudinal frame spacing = 2.50 transverse web spacing = 8.00
bottom modulus= 55350 deck modulus= 51702 estimated bare hull weight= 5140

longitudinal frame spacing = 2.50 transverse web spacing = 8.00
bottom modulus= 52240 deck modulus= 51678 estimated bare hull weight= 5080

longitudinal frame spacing = 2.50 transverse web spacing = 8.00
bottom modulus= 52568 deck modulus= 53466 estimated bare hull weight= 5132

longitudinal frame spacing = 2.50 transverse web spacing = 8.00
bottom modulus= 52568 deck modulus= 53466 estimated bare hull weight= 5132

longitudinal frame spacing = 2.50 transverse web spacing = 9.00
bottom modulus= 59735 deck modulus= 52807 estimated bare hull weight= 5191

longitudinal frame spacing = 2.50 transverse web spacing = 9.00
bottom modulus= 56874 deck modulus= 52358 estimated bare hull weight= 5116

longitudinal frame spacing = 2.50 transverse web spacing = 9.00
bottom modulus= 53535 deck modulus= 51799 estimated bare hull weight= 5042

longitudinal frame spacing = 2.50 transverse web spacing = 9.00
bottom modulus= 53535 deck modulus= 51799 estimated bare hull weight= 5042

longitudinal frame spacing = 2.50 transverse web spacing = 10.00
bottom modulus= 57730 deck modulus= 51366 estimated bare hull weight= 5130

longitudinal frame spacing = 2.50 transverse web spacing = 10.00
bottom modulus= 54725 deck modulus= 51317 estimated bare hull weight= 5076

FIGURE 3.9...(Continued)

```
longitudinal frame spacing =  2.50  transverse web spacing = 10.00
bottom modulus= 54976  deck modulus= 52444  estimated bare hull weight=  5111

longitudinal frame spacing =  2.50  transverse web spacing = 10.00
bottom modulus= 54976  deck modulus= 52444  estimated bare hull weight=  5111

longitudinal frame spacing =  2.75  transverse web spacing =  7.00
bottom modulus= 56646  deck modulus= 51234  estimated bare hull weight=  5325

longitudinal frame spacing =  2.75  transverse web spacing =  7.00
bottom modulus= 53737  deck modulus= 51237  estimated bare hull weight=  5245

longitudinal frame spacing =  2.75  transverse web spacing =  7.00
bottom modulus= 53737  deck modulus= 51237  estimated bare hull weight=  5245

longitudinal frame spacing =  2.75  transverse web spacing =  8.00
bottom modulus= 57897  deck modulus= 51889  estimated bare hull weight=  5218

longitudinal frame spacing =  2.75  transverse web spacing =  8.00
bottom modulus= 54807  deck modulus= 51405  estimated bare hull weight=  5136

longitudinal frame spacing =  2.75  transverse web spacing =  8.00
bottom modulus= 52031  deck modulus= 51375  estimated bare hull weight=  5086

longitudinal frame spacing =  2.75  transverse web spacing =  8.00
bottom modulus= 52458  deck modulus= 53721  estimated bare hull weight=  5151

longitudinal frame spacing =  2.75  transverse web spacing =  8.00
bottom modulus= 52458  deck modulus= 53721  estimated bare hull weight=  5151

longitudinal frame spacing =  2.75  transverse web spacing =  9.00
bottom modulus= 59115  deck modulus= 51281  estimated bare hull weight=  5178

longitudinal frame spacing =  2.75  transverse web spacing =  9.00
bottom modulus= 56185  deck modulus= 52003  estimated bare hull weight=  5127

longitudinal frame spacing =  2.75  transverse web spacing =  9.00
bottom modulus= 53279  deck modulus= 51498  estimated bare hull weight=  5081

longitudinal frame spacing =  2.75  transverse web spacing =  9.00
bottom modulus= 53279  deck modulus= 51498  estimated bare hull weight=  5081

longitudinal frame spacing =  2.75  transverse web spacing = 10.00
bottom modulus= 57320  deck modulus= 51356  estimated bare hull weight=  5110

longitudinal frame spacing =  2.75  transverse web spacing = 10.00
bottom modulus= 57320  deck modulus= 51356  estimated bare hull weight=  5110  bestlonspac= 2.250000000e+000 bestxvrspac=
9.000000000e+000   bestn=         2    smreq= 5.116718750e+004        bestmodbot= 5.403096746e+004  bestmoddk= 5.154864626e+004;
550   85.94 45.0 35.8 6.4 2.40 12.2 0.80  5016       539    5555 38685 33130110.08 42560
\014
```

FIGURE 3.9...(Continued)

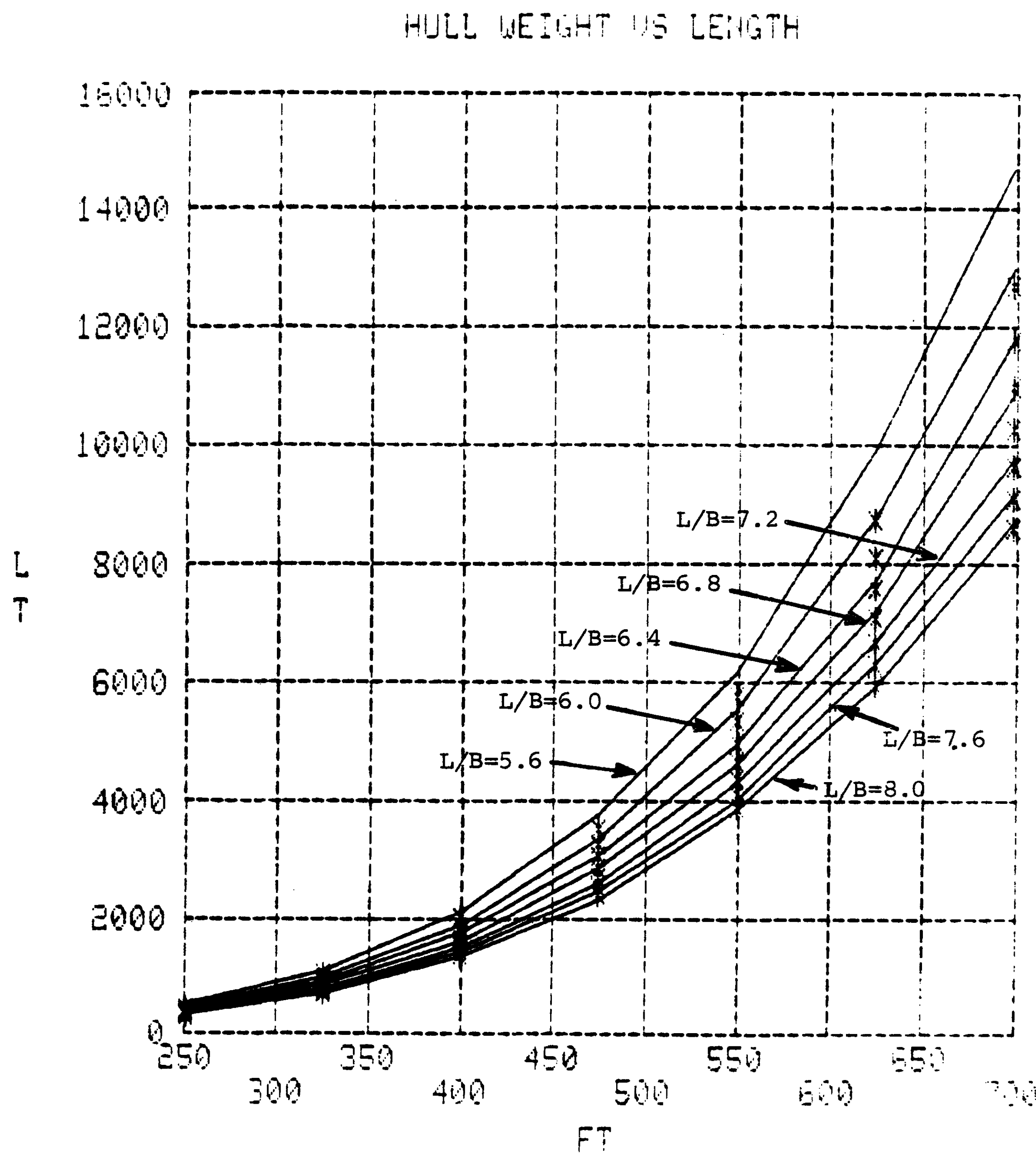


FIGURE 3.10

PLOT OF BARGE HULL STEEL WEIGHT VS. BARGE SCANTLING LENGTH

FOR C_B=0.75 AND B/T=2.2

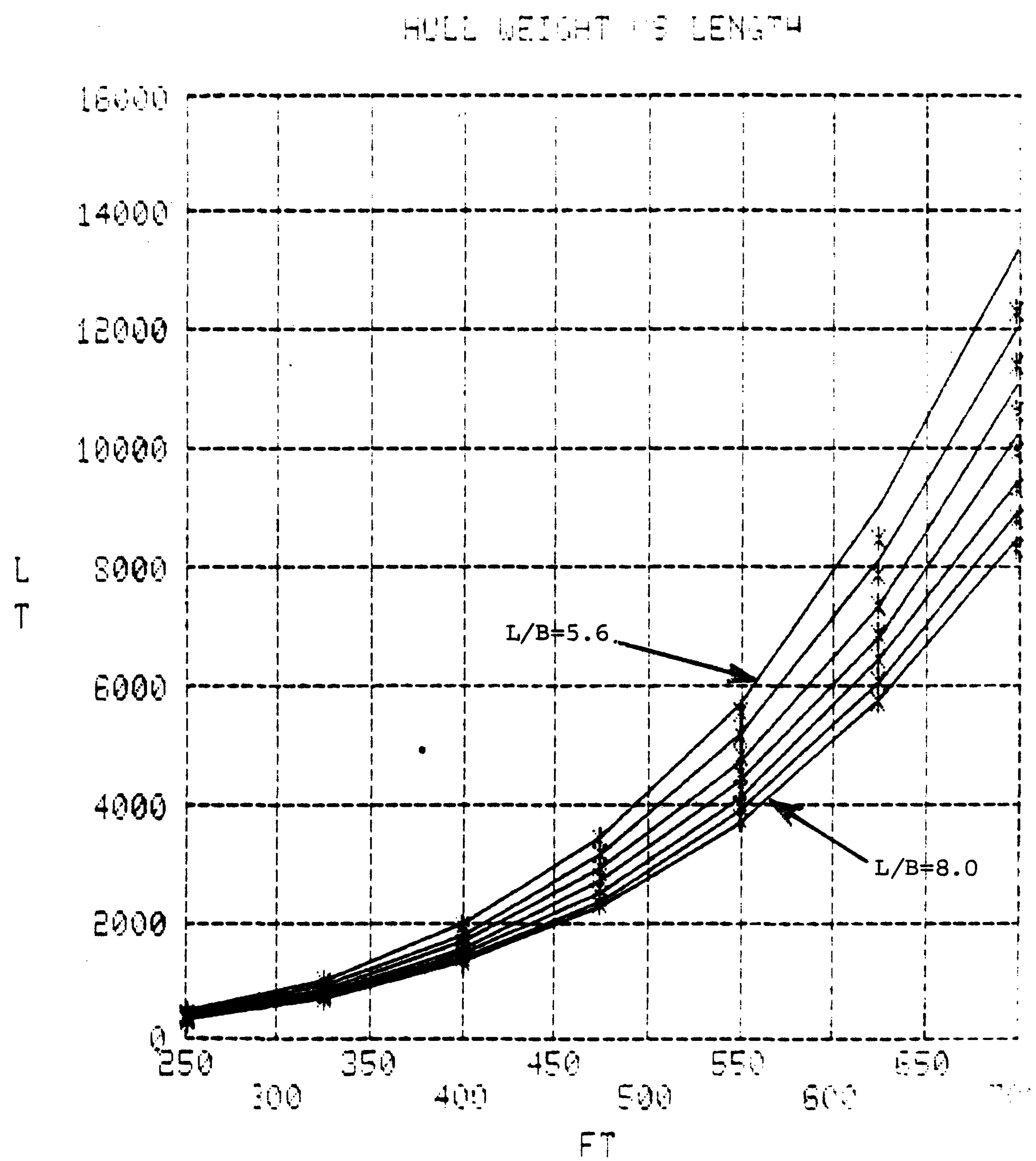


FIGURE 3.10

PLOT OF BARGE HULL STEEL WEIGHT VS. BARGE SCANTLING LENGTH

FOR C_B=0.75 AND B/T=2.4

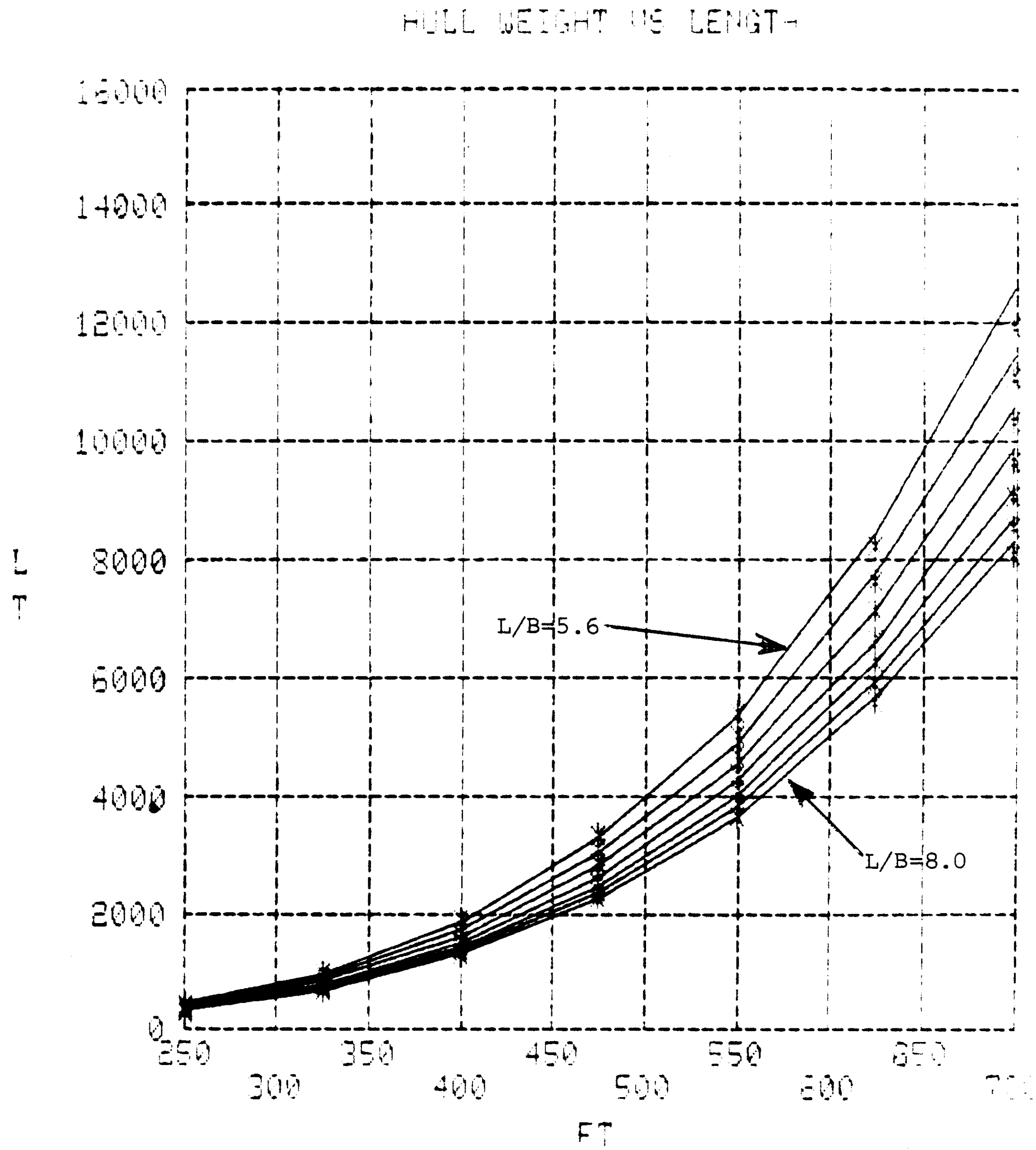


FIGURE 3.10

PLOT OF BARGE HULL STEEL WEIGHT VS. BARGE SCANTLING LENGTH

FOR C_B=0.75 AND B/T=2.6

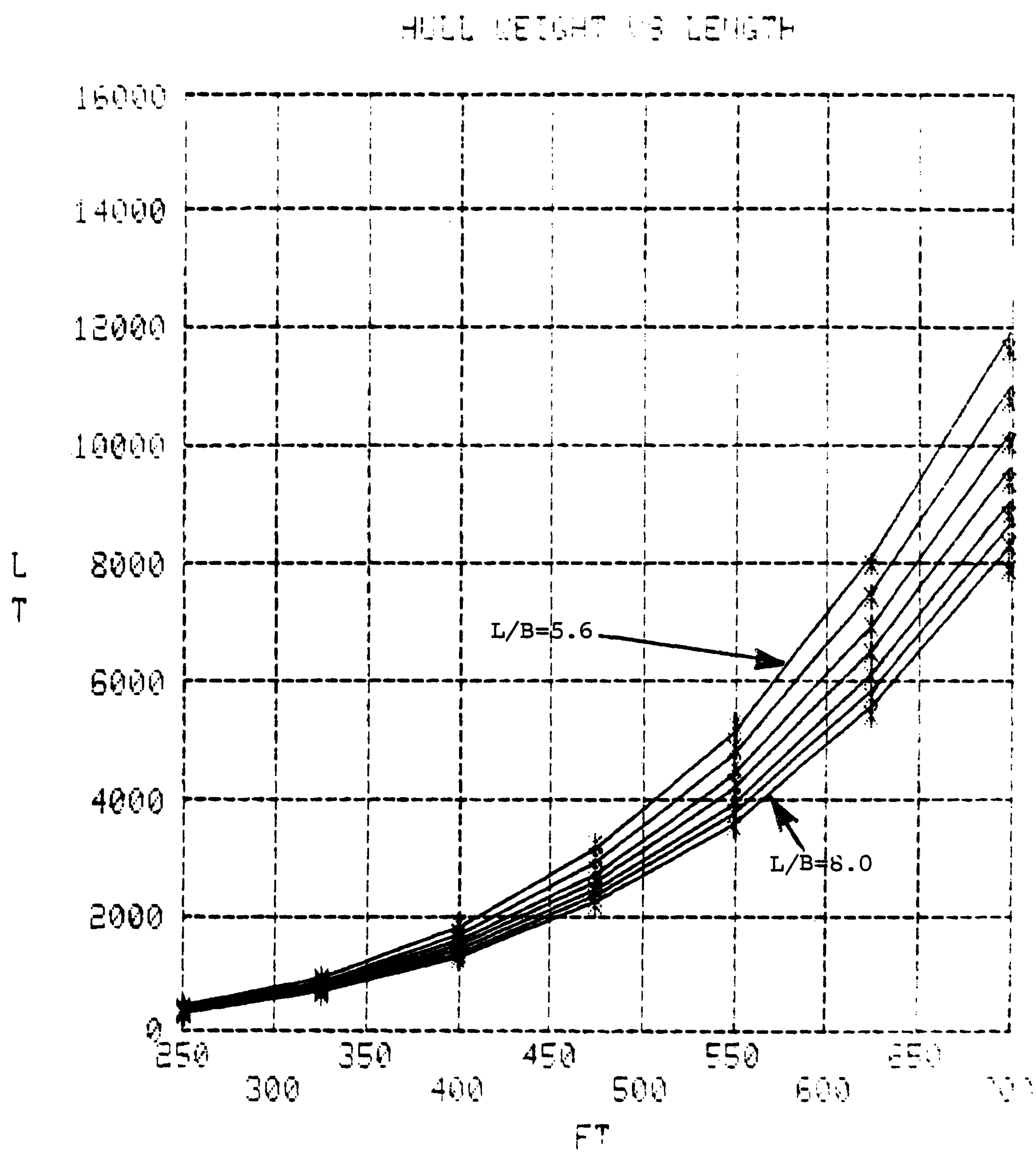


FIGURE 3.10

PLOT OF BARGE HULL STEEL WEIGHT VS. BARGE SCANTLING LENGTH

FOR C_B=0.75 AND B/T=2.8

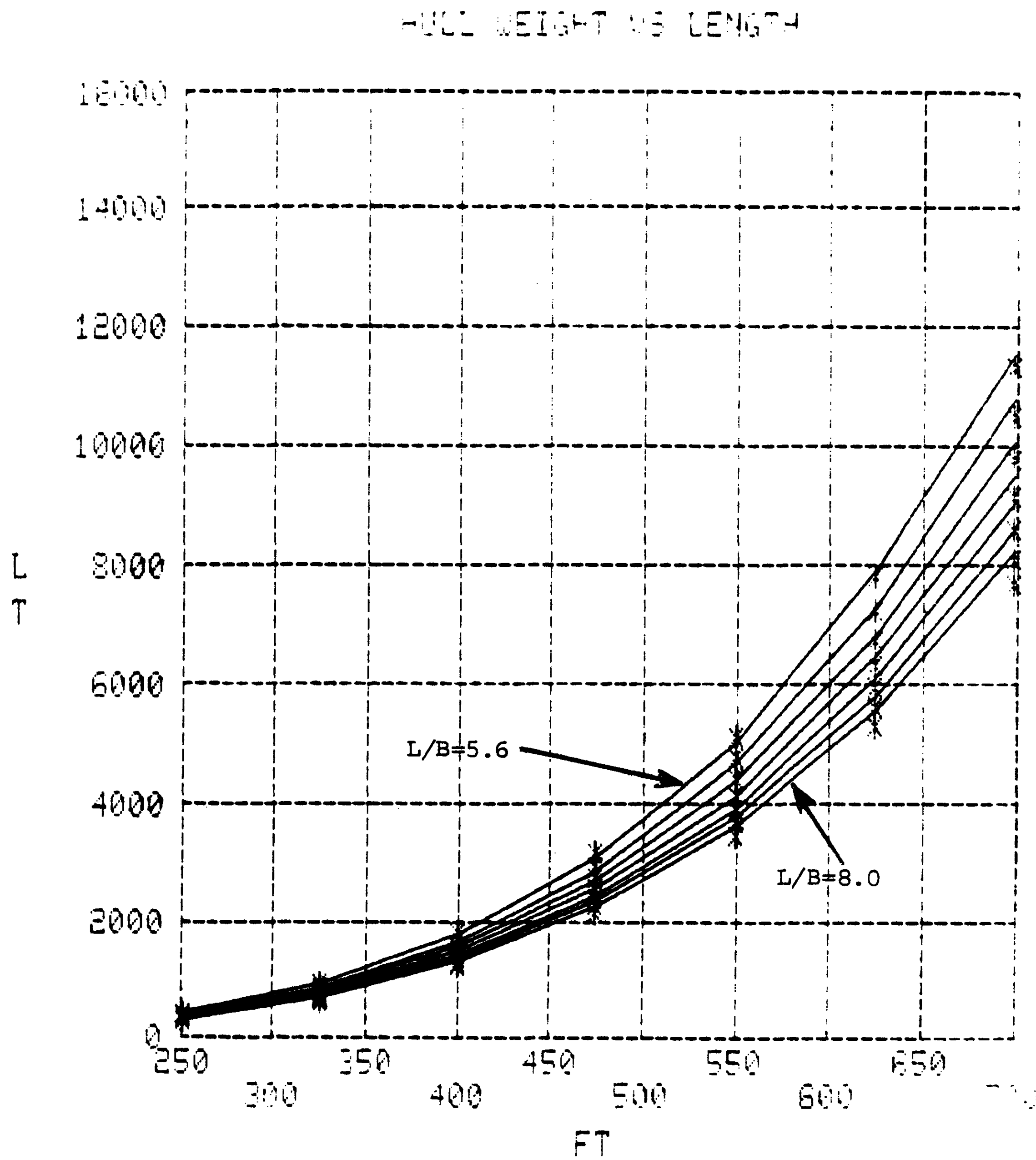


FIGURE 3.10

PLOT OF BARGE HULL STEEL WEIGHT VS. BARGE SCANTLING LENGTH

FOR C_B=0.75 AND B/T=3.0

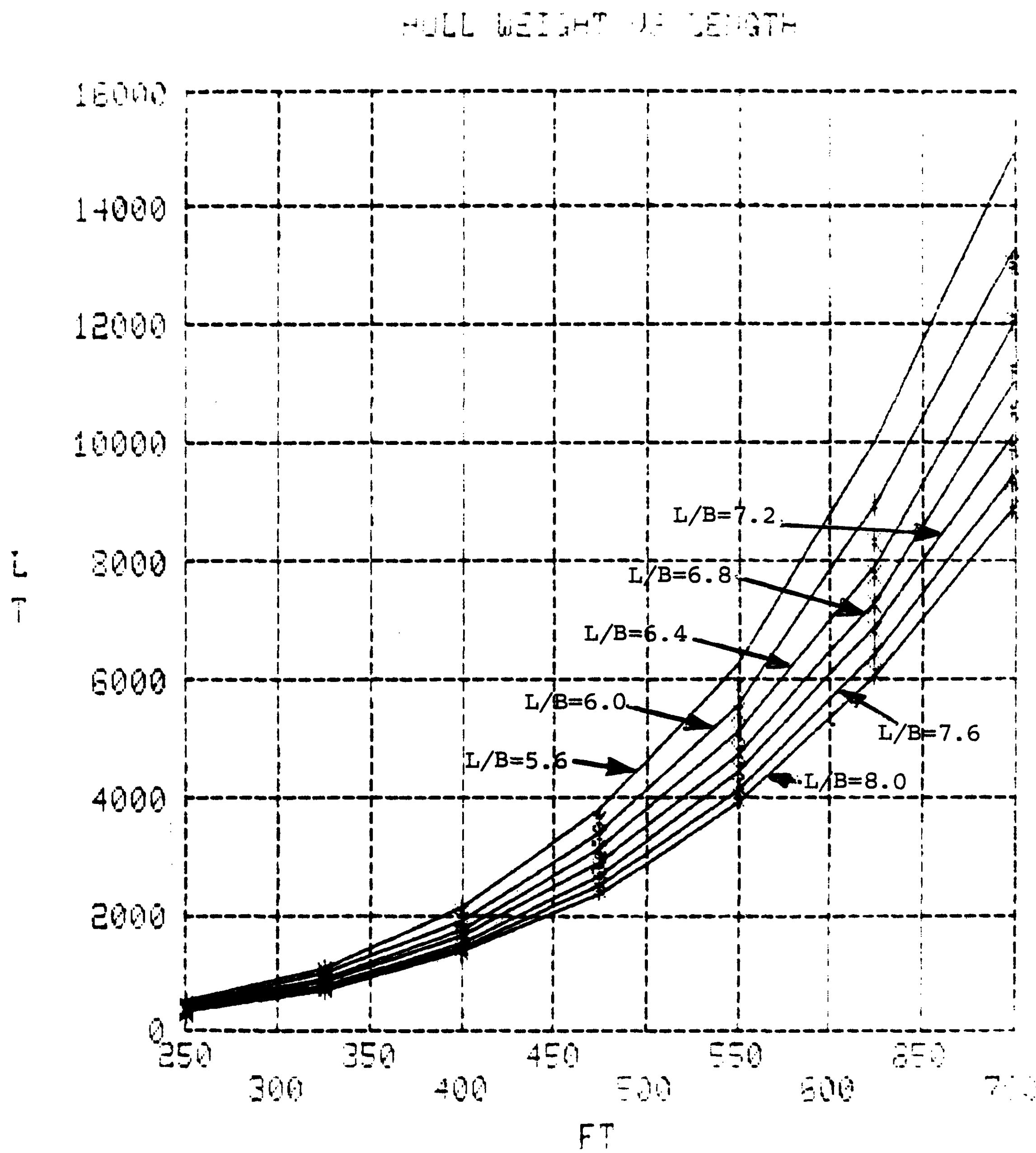


FIGURE 3.11

PLOT OF BARGE HULL STEEL WEIGHT VS. BARGE SCANTLING LENGTH

FOR C_B=0.80 AND B/T=2.2

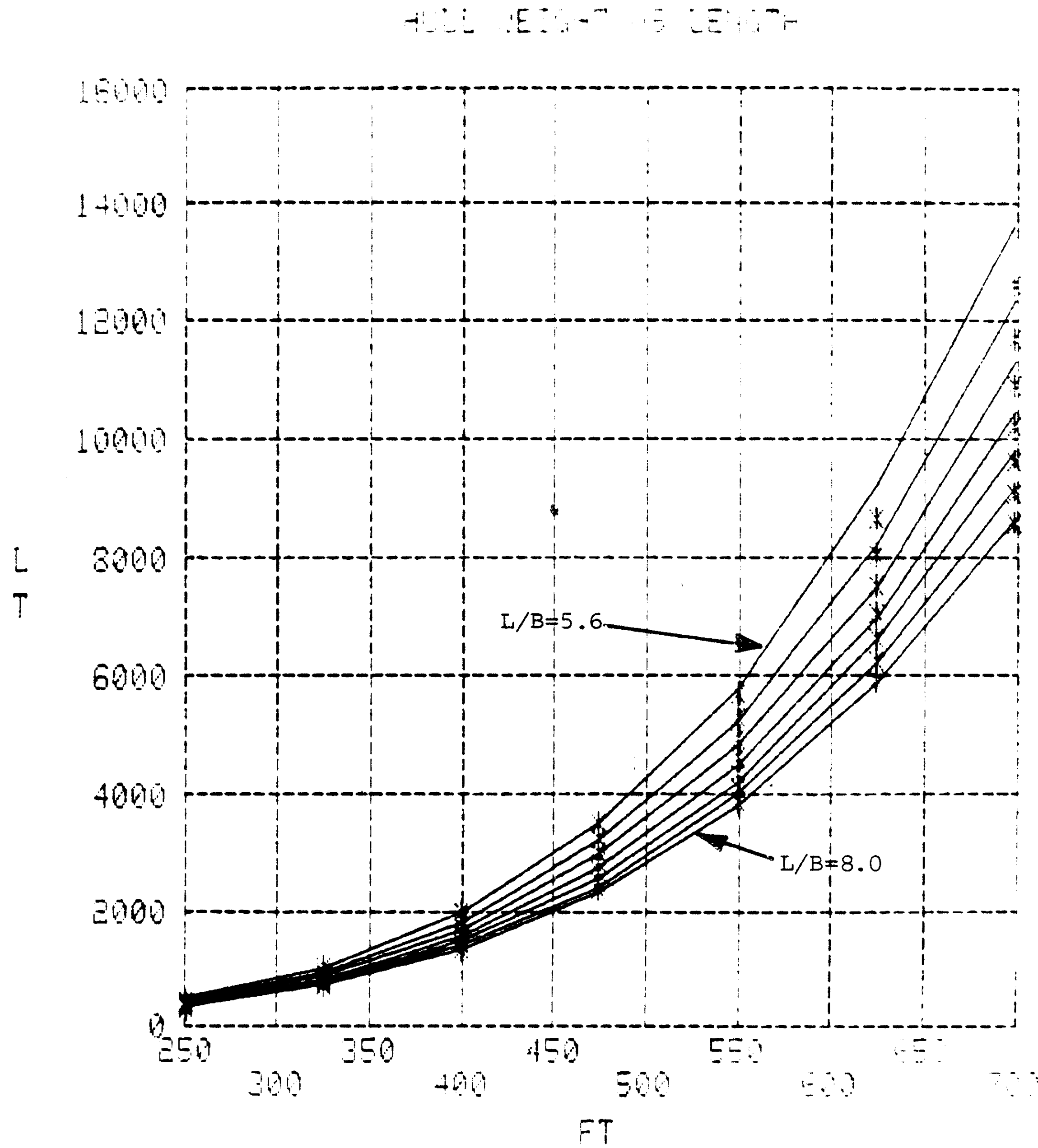


FIGURE 3.11

PLOT OF BARGE HULL STEEL WEIGHT VS. BARGE SCANTLING LENGTH

FOR C_B=0.80 AND B/T=2.4

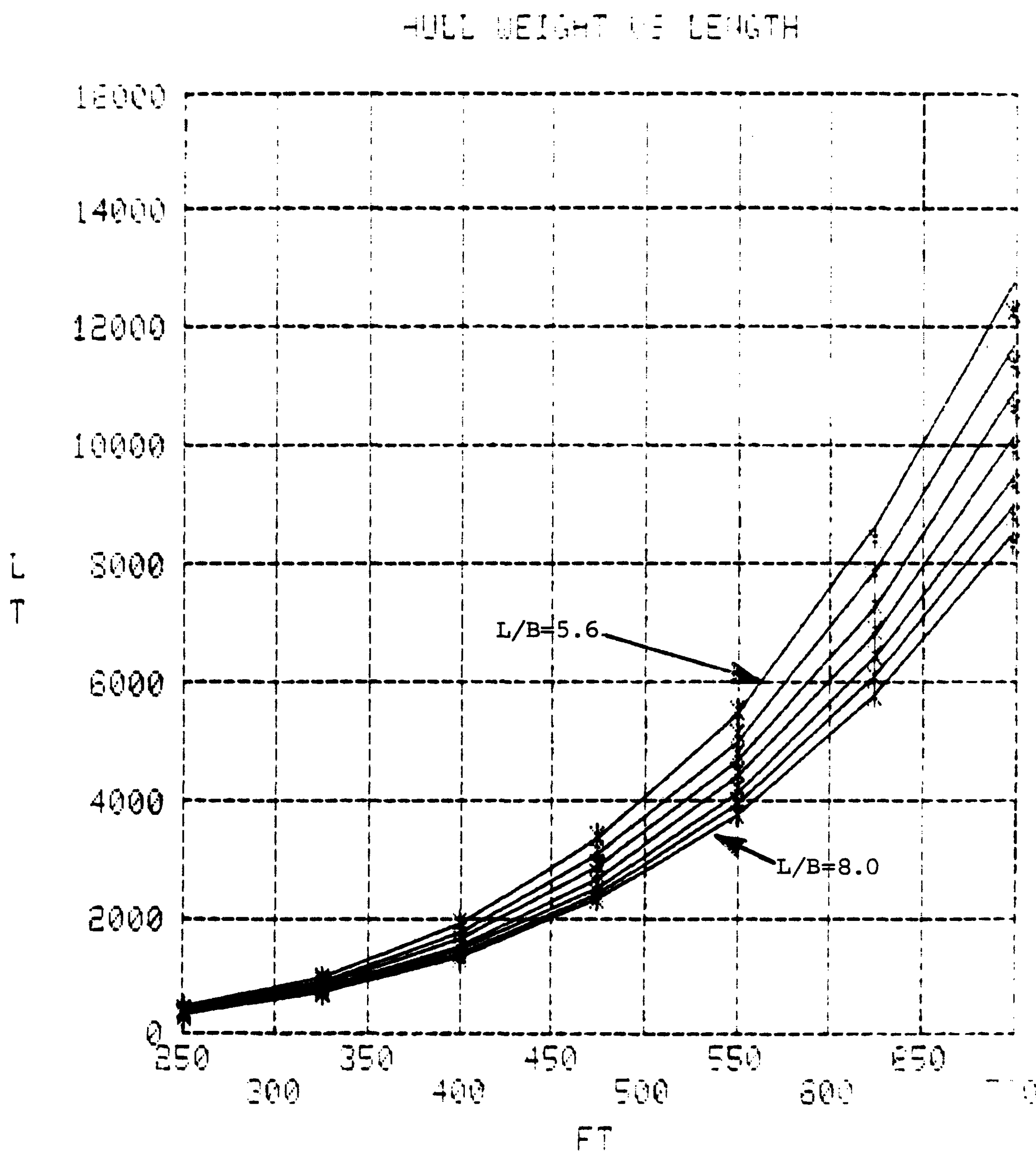


FIGURE 3.11

PLOT OF BARGE HULL STEEL WEIGHT VS. BARGE SCANTLING LENGTH

FOR C_B=0.80 AND B/T=2.6

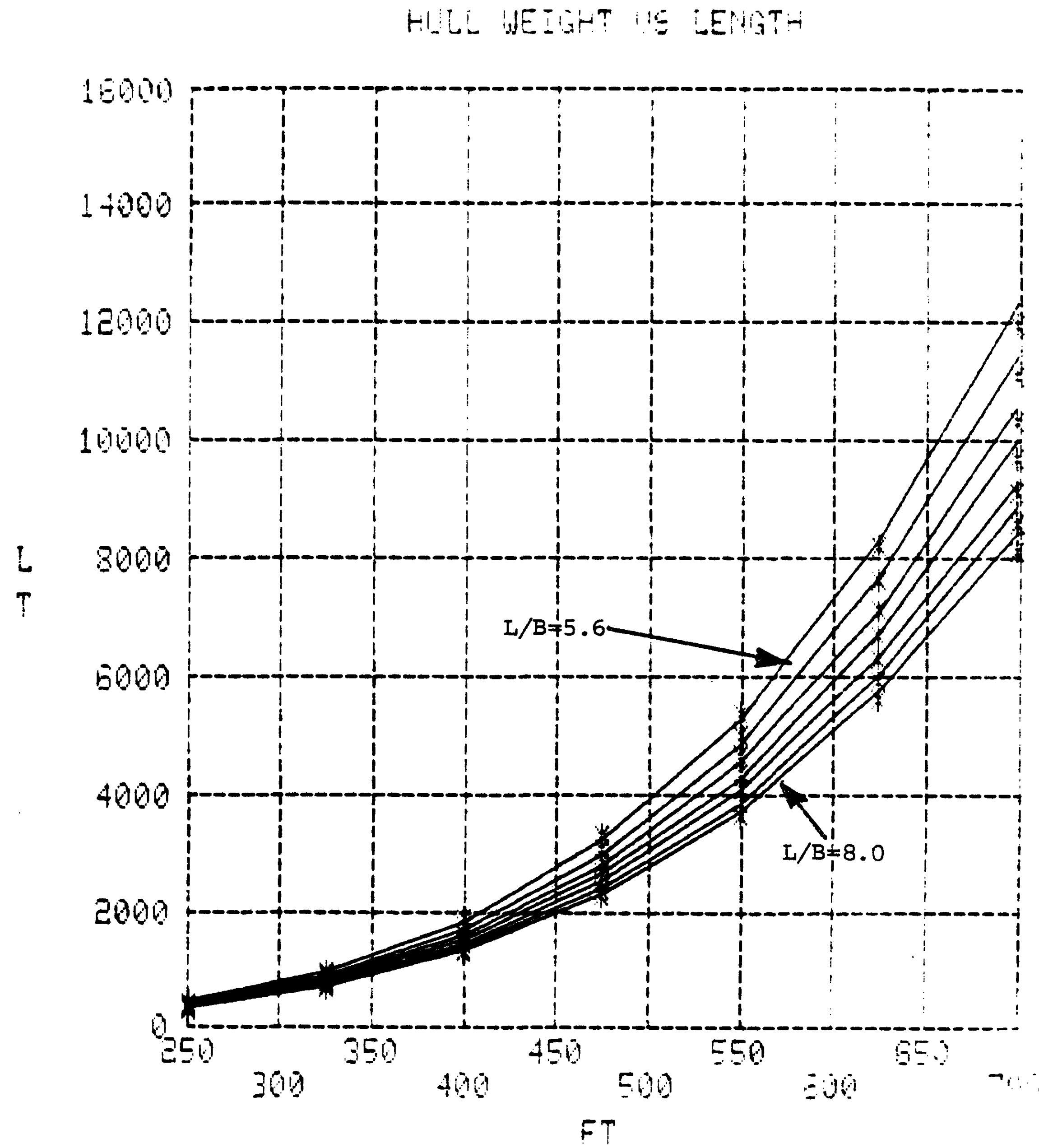


FIGURE 3.11

PLOT OF BARGE HULL STEEL WEIGHT VS. BARGE SCANTLING LENGTH

FOR C_B=0.80 AND B/T=2.8

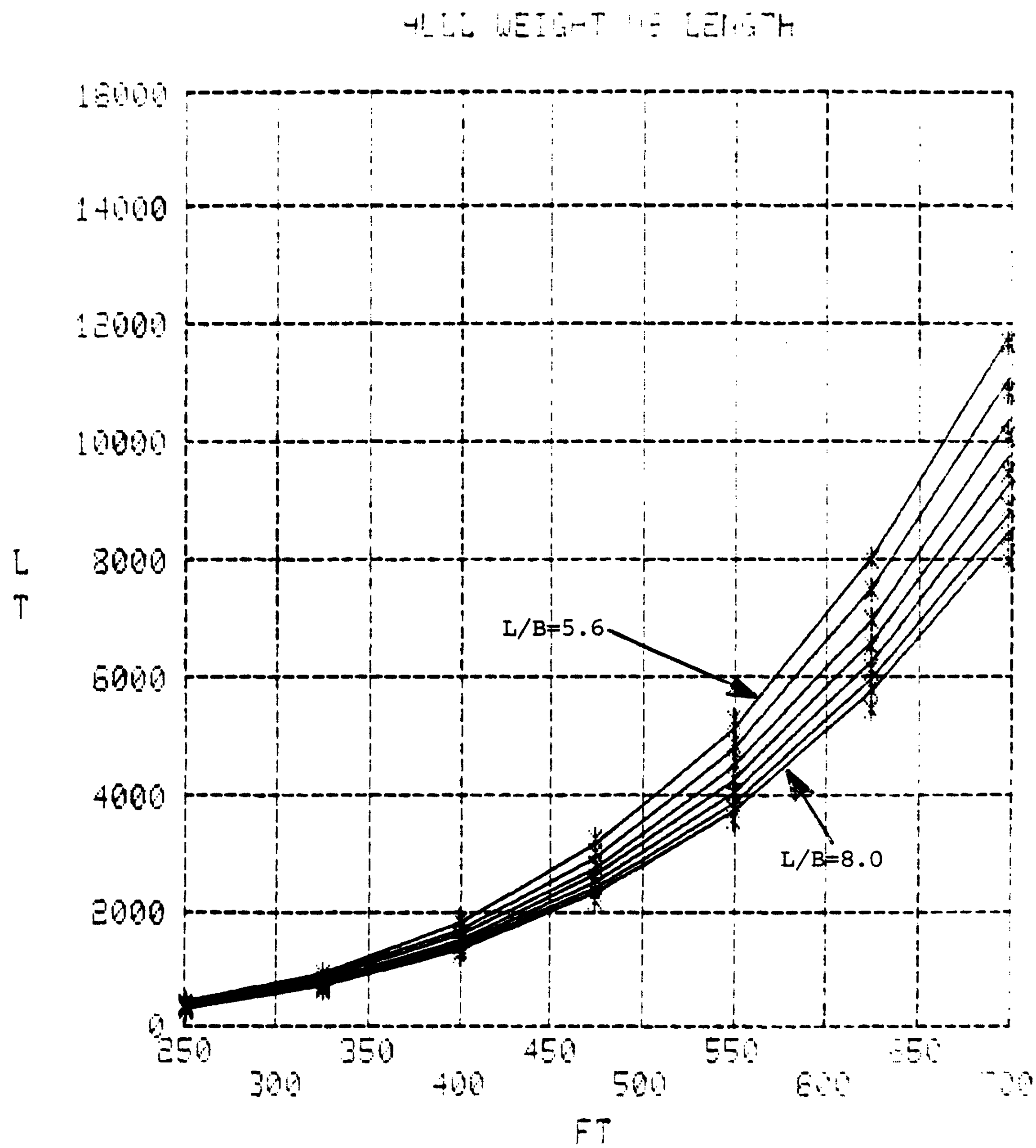


FIGURE 3.11

PLOT OF BARGE HULL STEEL WEIGHT VS. BARGE SCANTLING LENGTH

FOR C_B=0.80 AND B/T=3.0

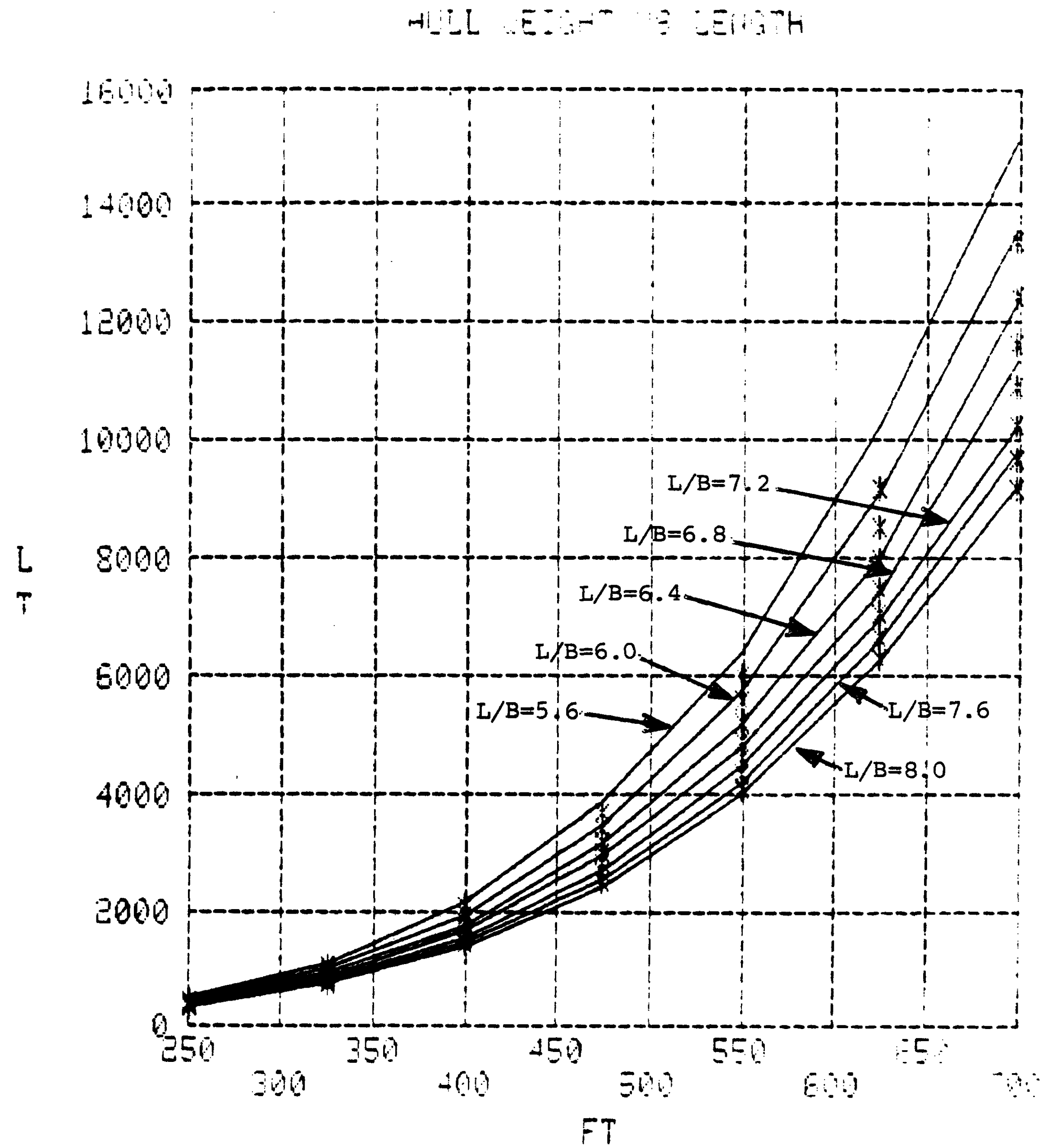


FIGURE 3.12

PLOT OF BARGE HULL STEEL WEIGHT VS. BARGE SCANTLING LENGTH

FOR C_B=0.85 AND B/T=2.2

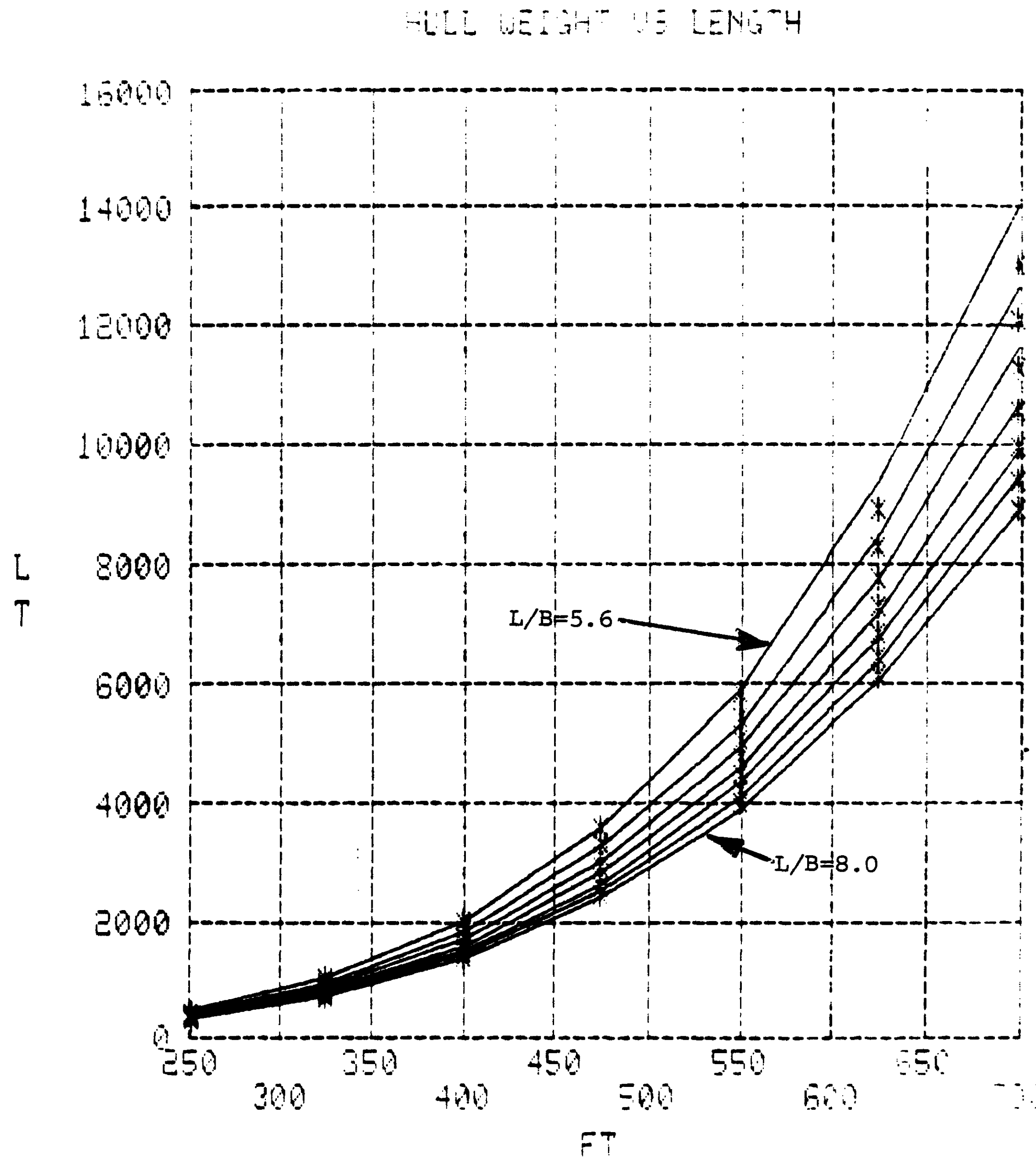


FIGURE 3.12

PLOT OF BARGE HULL STEEL WEIGHT VS. BARGE SCANTLING LENGTH

FOR C_B=0.85 AND B/T=2.4

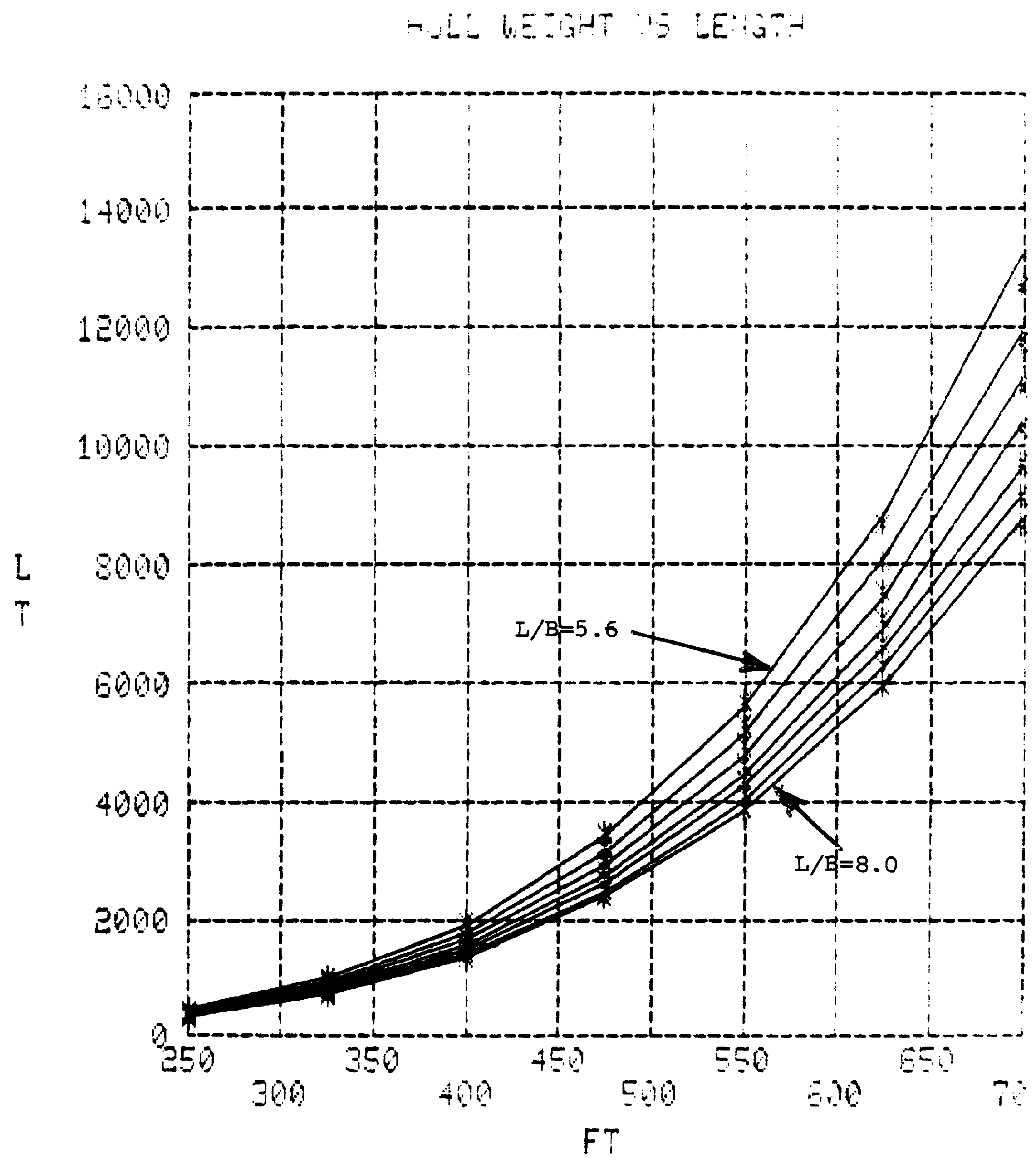


FIGURE 3.12

PLOT OF BARGE HULL STEEL WEIGHT VS. BARGE SCANTLING LENGTH

FOR C_B=0.85 AND B/T=2.6

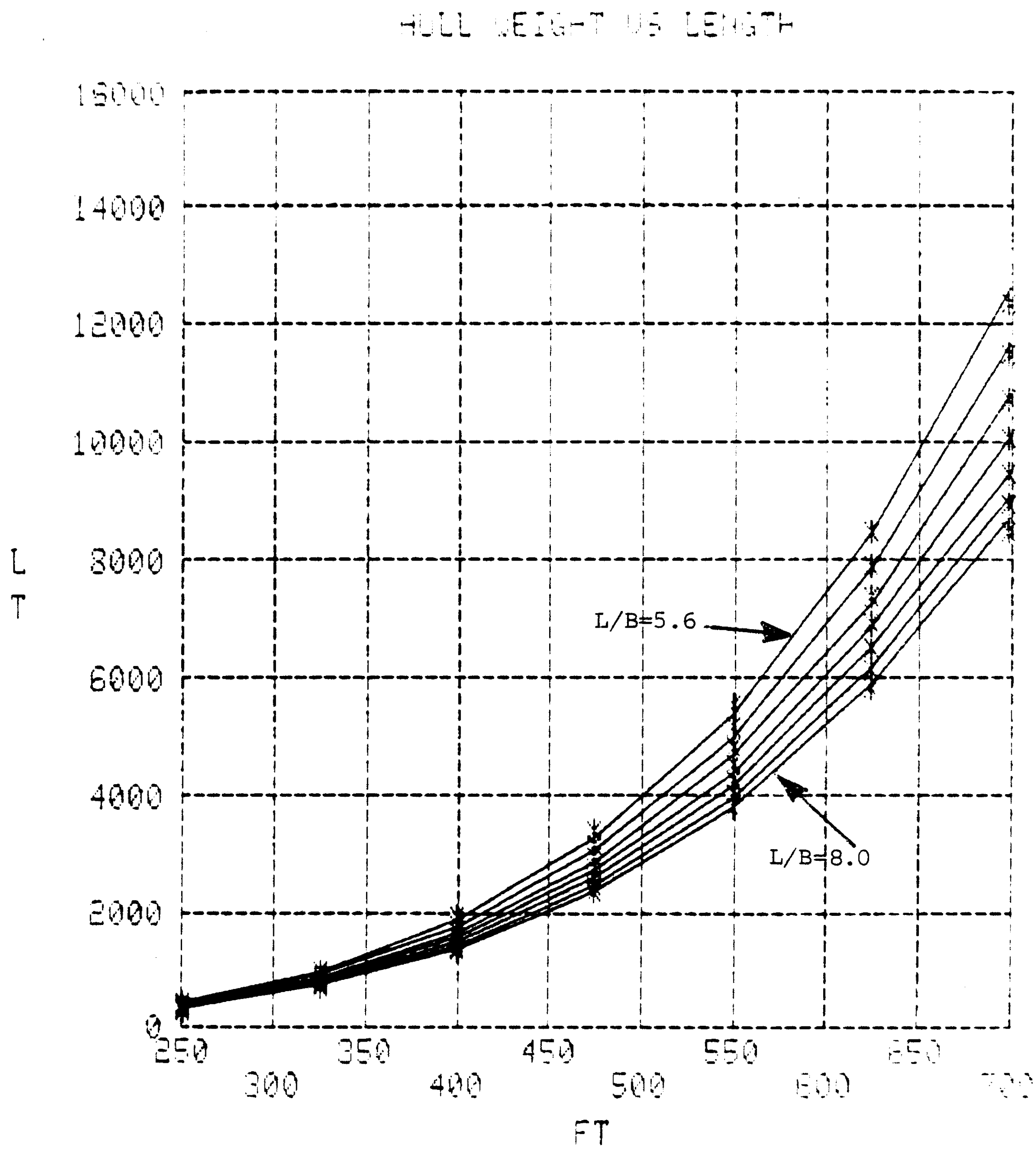


FIGURE 3.12

PLOT OF BARGE HULL STEEL WEIGHT VS. BARGE SCANTLING LENGTH

FOR C_B=0.85 AND B/T=2.8

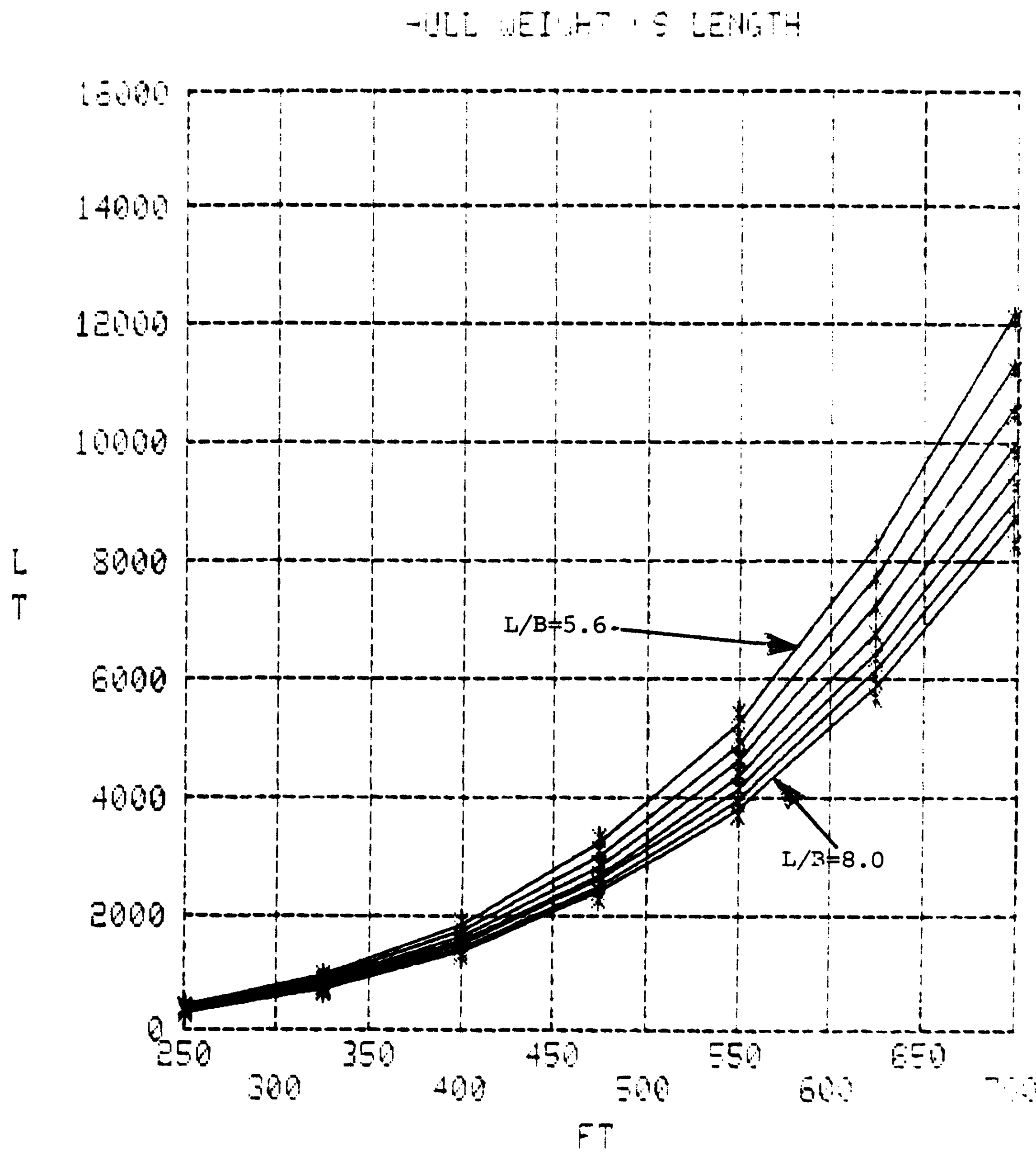


FIGURE 3.12

PLOT OF BARGE HULL STEEL WEIGHT VS. BARGE SCANTLING LENGTH

FOR C_B=0.85 AND B/T=3.0

TABLE 3.1

VALUES OF "FACT"

Barge Length (Meters)	"Fact"	Barge Length (Meters)	"Fact"
30	42.6	130	1661.0
35	56.0	135	1808.8
40	73.8	140	1961.6
45	95.4	145	2123.6
50	120.5	150	2292.9
55	151.9	155	2467.7
60	185.1	160	2650.8
65	228.4	165	2848.6
70	278.3	170	3049.7
75	338.0	175	3269.1
80	419.1	180	3504.6
85	540.4	185	3741.9
90	666.6	190	3993.9
95	787.2	195	4241.6
100	914.2	200	4499.4
105	1029.0	205	4767.3
110	1141.2	210	5058.9
115	1261.2	215	5348.8
120	1385.9	220	5640.9
125	1523.5	225	5953.2
		230	6269.5

TABLE 3.1...(Continued)

VALUES OF "FACT"

Barge Length (Feet)	"Fact"	Barge Length (Feet)	"Fact"	Barge Length (Feet)	"Fact"
100	6.8	320	132.0	540	439.0
110	8.1	330	144.0	550	458.0
120	9.3	340	155.0	560	477.0
130	11.2	350	165.0	570	498.0
140	13.1	360	176.0	580	519.0
150	15.3	370	186.0	590	542.0
160	17.5	380	199.0	600	564.0
170	20.4	390	210.0	610	587.0
180	23.4	400	223.0	620	611.0
190	26.5	410	236.0	630	635.0
200	29.7	420	249.0	640	658.0
210	34.0	430	262.0	650	682.0
220	38.3	440	276.0	660	707.0
230	43.3	450	291.0	670	732.0
240	48.2	460	305.0	680	759.0
250	55.1	470	320.0	690	787.0
260	62.0	480	336.0	700	814.0
270	74.0	490	352.0	710	842.0
280	85.0	500	368.0	720	869.0
290	97.0	510	385.0	730	899.0
300	109.0	520	402.0	740	928.0
310	120.0	530	420.0	750	958.0

Source: American Bureau of Shipping, Rule for Building and Classing Steel Barges for Offshore Service, 1973, Table 3.2

TABLE 3.2

DEPTH OF LONGITUDINAL BULKHEAD AND SIDE GIRDERS

AND TOTAL NUMBER OF STANCHIONS INSERTED BETWEEN SIDES

"b" Barge Molded Breadth	Depth of Longitudinal Bulkhead and Side Girders	Total No. of Stanchions Inserted Between Sides
< 35'	5'	0
35'- 67'	7'	2
67'- 99'	9'	4
> 99'	11'	6

TABLE 3.3

MINIMUM PLATE THICKNESS

Plate Type	Applicable Rule	Minimum Thickness Calculation (mm and inches)	Valid Range for Equation
Bottom	ABS(1973) para.4.3.3	"botplt"=0.0463x"l"+0.007xs+1.7mm	"l"≤ 122m
		=0.0833x"l"+0.01xs-5.1 mm	122<"l"≤ 228.5m
		"botplt"=0.000555x"l"+0.007xs+0.067 in	"l"≤ 400 ft
		=0.001x"l"+0.01 xs-0.2 in	400 <"l"≤ 750 ft
Side	ABS(1973) para 4.3.1	"sidplt"=0.06347x"l"+0.0075xsmm	"l"≤ 152.5m
		=0.02706x"l"+0.01xs +3.86mm	152.5<"l"≤228.5m
		"sidplt"=0.00082x"l"+0.0075xs,in	"l"≤500 ft
		=0.00035x"l"+0.01xs+0.152 in	500<"l"≤ 750 ft
Deck	ABS(1973) para 5.1.1	"dkplt" =0.01xs + 1.8mm	s ≤ 711mm
		=0.00643xs+4.3mm	711 < s< 1016mm
		"dkplt" = 0.01xs+0.07 in	s ≤ 28 in
		= 0.00643xs+0.17 in	28 < s < 40 in
Longitudinal Bulkhead	ABS(1973) para 9.9.1	"bhdplt"= $\frac{s\sqrt{h}}{254}$ + 2.5mm	h="d"+1.22-"dbdep"
		"bhdplt"= $\frac{s\sqrt{h}}{460}$ + 0.10in	h="d"+4-"dbdep"

Notes: s: longitudinal frame spacing (12x"lonspac")

h: bulkhead plate head; the distance from the lower edge of longitudinal bulkhead plating to 4' (1.22m) above the deck

TABLE 3.4

MINIMUM REQUIRED SECTION MODULI

FOR LONGITUDINAL FRAMES

Longitudinal frames must have section moduli "secmod$\underline{x}$"
greater than the value obtained from the formulae:

$$\text{"secmod}\underline{x}\text{"} = 7.9 \times c \times h \times s \times l^2 \ \text{cm}^3$$

or

$$\text{"secmod}\underline{x}\text{""} = 0.0041 \times c \times h \times s \times l^2 \ \text{in}^3$$

where the appropriate values of x,c,h,s and l are found in the table below:

Frame Type	Applicable Rule	x	c	h*	s	l
Bottom	ABS(1973) para 6.15.1	1	1.34 1.30***	"h1"	"lonspac"	"xvrspac"
Side	ABS(1973) para 6.15.1	3	1.25	"h2"**	"	"
Deck	ABS(1973) para 6.15.1	5	1.25	"h3"	"	"
Longitudinal Bulkhead	ABS(1973) para 9.9.2	7	1.00	"h2"**	"	"
Double-Bottom	ABS(1973) para 6.19.2	9	1.00	"h5"	"	"

*h: head of the associated member as shown in Figures 3.1 and 3.2

**"h2": Average head experienced by side/bulkhead members. It was used since the required and calculated section moduli are both linear with respect to head

***: C-value for bottom longitudinals when double bottom is installed

TABLE 3.5

MINIMUM REQUIRED SECTION MODULI
FOR TRANSVERSE GIRDERS

Transverse girders must have section moduli "secmod$\underline{x}$" greater than the value obtained from the formulae:

"secmod$\underline{x}$" $= 4.74 \times c \times h \times s \times l^2$ cm^3

or

"secmod$\underline{x}$" $= 0.0025 \times c \times h \times s \times l^2$ in^3

where the appropriate values of x,c,h,s and l are found in the table below:

Girder Type	Applicable Rule	x	c	h*	s	l**
Bottom	ABS(1973) para 6.15.2	2	1.75	"hl"	"xvrspac"	"12"
Side	ABS(1973) para 6.15.2	4	1.75	"h2"	"	"14"
Deck	ABS(1973) para 6.15.2	6	2.00	"h3"	"	"12"
Longitudinal Bulkhead	ABS(1973) para 9.9.3	8	1.50	"h2"	"	"14"

*h: head of the associated member as shown in Figures 3.1 and 3.2

**"l": the unsupported span of the girder as previously calculated in Section 3.2.3 and shown in Figures 3.1 and 3.2

TABLE 3.6

TRANSVERSE MEMBER WEIGHT PER LONGITUDINAL FOOT CALCULATIONS

Member	Weight Per Longitudinal Foot Program Variable = [Steel Volume] x [Longitudinal Weight Per Foot Factor - $\text{lb/in}^2\text{-ft}$]
Bottom Transverse Girder	$\text{"xvrs1"} = \left[\begin{pmatrix}\text{cross-sectional area*}\\ \text{of member-"area "}\end{pmatrix} \times \text{"b"}\right] \times \left(\frac{3.4}{\text{"xvrspac"}}\right)$
Side Transverse Girders(2)	$\text{"xvrs2"} = \left[2 \times \begin{pmatrix}\text{cross sectional area*}\\ \text{of member-"area4"}\end{pmatrix} \times \left(\text{"d"} - \frac{\text{"dbdep"}}{12}\right)\right] \times \left(\frac{3.4}{\text{"xvrspac"}}\right)$
Deck Transverse Girder	$\text{"xvrs3"} = \left[\begin{pmatrix}\text{cross sectional area*}\\ \text{of member-"area6"}\end{pmatrix} \times \text{"b"}\right] \times \left(\frac{3.4}{\text{"xvrspac"}}\right)$
Longitudinal Bulkhead Transverse Girder	$\text{"xvrs4"} = \left[\begin{pmatrix}\text{cross sectional area*}\\ \text{of member-"area6"}\end{pmatrix} \times \left(\text{"d"} - \frac{\text{"dbdep"}}{12}\right)\right] \times \left(\frac{3.4}{\text{"xvrspac"}}\right)$
Transverse Bulkhead Plating	$\text{"xvrs5"} = \left[\frac{1}{\text{"ntrbkd"}} \times \left(\text{"b"} \times \left(\text{"d"} - \frac{\text{"dbdep"}}{12}\right) \times \text{"bhdplt"}\right)\right] \times \left(\frac{3.4 \times 12}{\text{"xvrspac"}}\right)$
Transverse Bulkhead Longitudinal Stiffeners	$\text{"xvrs6"} = \left[\frac{1}{\text{"ntrbkd"**}} \times \left(\text{"b"} \times \left(\frac{d - \frac{\text{"dbdep"}}{12}}{\text{"lonspac"***}}\right) \times \begin{pmatrix}\text{cross sectional area}\\ \text{of member-"area7"*}\end{pmatrix}\right)\right] \times \left(\frac{3.4}{\text{"xvrspac"}}\right)$
Stanchions	$\text{"xvrs7"} = \left[\begin{pmatrix}\text{number of stanchions}\\ \text{per transverse girder}\\ \text{"stans"}\end{pmatrix} \times \begin{pmatrix}\text{cross sectional area}\\ \text{of stanchion"astans"}\end{pmatrix} \times \begin{pmatrix}\text{length of}\\ \text{stanchion}\end{pmatrix}\right] \times \left(\frac{3.4}{\text{"xvrspac"}}\right)$ where $\text{"astan"} = \pi \times \begin{pmatrix}\text{radius of the}\\ \text{cylindrical shaped}\\ \text{stanchion****}\end{pmatrix}^2$

* Cross sectional areas of members are obtained from scantlings calculated in "smact2".

** "ntrbkd" is the number of transverse girders between each transverse bulkhead.

*** $\left(\frac{d - \frac{\text{"dbdep"}}{12}}{\text{"lonspac"}}\right)$ is the number of stiffeners attached to the transverse bulkhead plate.

**** The radius of the stanchion was determined by equating the minimum stanchion load equations found in ABS (1973, par. 6.7.2 and Figure (6)). Specifically,

$$\begin{Bmatrix}1.07\\ 0.03\end{Bmatrix} \times \text{"12"} \times \text{"h"} \times \text{"xvrspac"} = w = \begin{Bmatrix}[1.232 - 0.00452\ (\text{"14"}/r)]\pi r^2 & (MT)\\ [7.83 - 0.345\ (\text{"14"}/r)]\pi r^2 & (LT)\end{Bmatrix}$$

where r is radius of stanchion and "12", "14", "h", "xvrspac" are as shown in Figures 3.1 and 3.2.

TABLE 3.7

INITIAL SCANTLING VALUES USED IN SUBROUTINE "smact2"

Dimension	Reference	Initial Value
Plate thickness		as specified in input "tp"
Plate depth		as specified in input "s"
Web depth	ABS (1973) para 9.9.3c ABS (1978) para 22.27.5	the larger of (1) 2.5 x long. frame and: (2) 1.5 x "14", for side and bulkhead girders 2.4 x "12", for bottom girders 1.5 x "12", for deck girders and then rounded up to nearest 1/4"
Web thickness	ABS (1978) Table 22.3 ABS (1973) para 9.9.3c reasonable assumption	the larger of: (1) 0.34" for "1" < 200' 0.34" + ("1" - 200)/3500 for 200' < "1" < 270' 0.36" + ("1" - 270)/3000 for 270' < "1" < 700 0.50" for 700' < "1" for deck, bottom, and side girders or the smaller of: (a) 0.44" and (b) 0.12" + $\frac{\text{"webdepth"}}{100}$ for longitudinal bulkhead girder and: (2) plate thickness for plate thicker than 5/8" or plate thickenss less 1/4" for plate thinner than 5/8" and then rounded up to nearest 1/16"
Flange width	reasonable minimum Horne (1958)	the smaller of: (1) web depth/5 and: (2) 20 x web thickness rounded up to the nearest 1/4"
Flange thickness	reasonable assumption	plate thickness

TABLE 3.8

DOUBLE BOTTOM PARTICULARS

Member	Applicable Rule	Minimum Dimensions (Metric and English units)
Double-Bottom Depth	ABS (1978) para 7.3.2	"dbdep" = 0.384 x "b" + 4.13$\sqrt{\text{"t"}}$ m = 32 x "b" + 190 $\sqrt{\text{"t"}}$ ft
Double-Bottom Plate Thickness	ABS (1978) para 7.5	"dbplt" = 0.037 x "l" + 0.009 x s + 0.5 mm = 0.000445 x "l"+ 0.009 x s +0.02 in where s = 12 x "lonspac"
Double-Bottom Center Girder Plate Thickness	ABS (1978) para 7.3.2	"cendbgirdth" = 0.056 x "l" + 5.5 mm = 0.00067 x "l" + 0.22 in
Double-Bottom Side Girder and Floor Plate Thickness	ABS (1978) para 7.3.4.	"sidbgirdth" = 0.036 x "l" + 4.7 mm = 0.00043 x "l" + 0.24 in
Double-Bottom Head	ABS (1973) para 6.19.2	"h5" = "d"-"dbdep"+ $\begin{cases} 1.2192 & \text{"l"} < 60.96\text{m} \\ \text{"l"}/50 & 60.96 \le \text{"l"} \le 121.92 \\ 2.4384 & 121.92 < \text{"l"} \end{cases}$ (meters) "h5" = "d"-"dbdep"+ $\begin{cases} 4 & \text{"l"} < 200 \\ \text{"l"}/50 & 200 \le \text{"l"} \le 400 \\ 8 & 400 < \text{"l"} \end{cases}$ (feet)
Double-Bottom Longitudinal Frame Section Modulus	ABS (1973) para 6.19.2	max ("secmod9"*, 0.85 x "secmodl"*)

*"secmod9" and "secmodl" values determined from expressions found in Table 3.4

TABLE 3.9

SINGLE-SKIN TANK BARGE HULL WEIGHTS FOR CB = 0.75

B/T	L/B	BARGE LENGTH (FT)						
		250	325	400	475	550	625	700
2.2	5.6	472	1061	2107	3763	6156	9946	14723
	5.8	452	992	1987	3542	5768	9317	13771
	6.0	431	942	1873	3357	5524	8785	12994
	6.2	412	907	1785	3193	5214	8502	12368
	6.4	396	868	1735	3057	4956	7736	11811
	6.6	380	829	1651	2957	4787	7458	11228
	6.8	364	802	1584	2848	4600	7194	10883
	7.0	351	771	1519	2730	4428	6903	10353
	7.2	360	754	1478	2592	4289	6691	9796
	7.4	346	728	1446	2521	4178	6457	9492
	7.6	335	704	1395	2456	4007	6302	9188
	7.8	321	689	1377	2364	3914	6106	8954
	8.0	307	672	1337	2298	3825	5906	8705
2.4	5.6	447	983	1960	3443	5660	9056	13358
	5.8	427	936	1861	3295	5345	8542	12774
	6.0	406	891	1774	3164	5188	8144	12021
	6.2	389	858	1701	3030	4918	7762	11570
	6.4	374	818	1640	2907	4717	7337	11062
	6.6	360	800	1592	2810	4581	7113	10633
	6.8	347	769	1526	2701	4382	6808	10235
	7.0	335	748	1482	2628	4239	6678	9937
	7.2	344	727	1450	2513	4106	6429	9441
	7.4	329	704	1402	2439	4003	6226	9145
	7.6	316	686	1376	2349	3892	6056	8929
	7.8	302	676	1328	2318	3779	5888	8710
	8.0	293	660	1304	2275	3696	5762	8469

TABLE 3.9 --Continued

SINGLE-SKIN TANK BARGE HULL WEIGHTS FOR CB = 0.75

B/T	L/B	BARGE LENGTH (FT)						
		250	325	400	475	550	625	700
	5.6	423	937	1853	3289	5345	8461	12591
	5.8	406	895	1792	3160	5090	8075	11968
	6.0	389	861	1714	3026	4860	7778	11422
	6.2	372	831	1649	2900	4703	7419	10952
	6.4	358	801	1595	2804	4521	7145	10584
	6.6	346	776	1551	2725	4404	6892	10224
2.6	6.8	327	753	1494	2618	4260	6615	9890
	7.0	315	738	1460	2530	4121	6433	9607
	7.2	326	717	1417	2458	4008	6275	9230
	7.4	315	706	1378	2412	3900	6091	8993
	7.6	304	·687	1345	2347	3800	5930	8728
	7.8	291	669	1305	2290	3735	5770	8519
	8.0	282	653	1289	2253	3631	5663	8323
	5.6	407	908	1795	3158	5138	8129	11909
	5.8	390	872	1732	3050	4917	7852	11432
	6.0	374	835	1678	2919	4755	7512	11018
	6.2	358	813	1619	2809	4581	7257	10601
	6.4	338	796	1572	2705	4423	6933	10238
	6.6	325	768	1502	2647	4304	6678	9991
2.8	6.8	313	759	1490	2572	4165	6518	9641
	7.0	302	733	1433	2515	4055	6337	9378
	7.2	316	712	1406	2443	3930	6128	9079
	7.4	305	693	1368	2408	3815	6001	8857
	7.6	295	699	1333	2343	3742	5847	8664
	7.8	282	686	1314	2287	3651	5696	8453
	8.0	273	666	1273	2258	3590	5573	8280

TABLE 3.9 --Continued

SINGLE-SKIN TANK BARGE HULL WEIGHTS FOR CB = 0.75

B/T	L/B	BARGE LENGTH (FT)						
		250	325	400	475	550	625	700
	5.6	395	887	1761	3096	4981	7908	11541
	5.8	370	854	1699	2944	4822	7592	11100
	6.0	354	835	1634	2853	4657	7289	10794
	6.2	339	803	1594	2746	4515	7099	10429
	6.4	326	796	1545	2698	4366	6810	10097
	6.6	314	768	1501	2607	4230	6604	9799
3.0	6.8	302	745	1472	2551	4112	6460	9503
	7.0	289	750	1430	2487	4021	6300	9298
	7.2	308	736	1393	2433	3897	6086	9045
	7.4	298	711	1355	2405	3828	5944	8760
	7.6	291	694	1345	2347	3752	5796	8611
	7.8	281	684	1316	2314	3688	5667	8424
	8.0	275	669	1299	2258	3617	5556	8253

TABLE 3.10

SINGLE-SKIN TANK BARGE HULL WEIGHTS FOR CB = 0.80

B/T	L/B	BARGE LENGTH (FT)						
		250	325	400	475	550	625	700
2.2	5.6	473	1064	2122	3793	6223	10076	14948
	5.8	452	996	2012	3630	6016	9666	14434
	6.0	432	961	1894	3384	5547	8976	13262
	6.2	413	909	1820	3250	5317	8530	12652
	6.4	397	870	1737	3138	5115	7917	12002
	6.6	382	835	1669	3003	4875	7558	11620
	6.8	365	809	1624	2893	4702	7307	11039
	7.0	352	786	1549	2789	4512	7039	10701
	7.2	361	757	1505	2645	4404	6883	10120
	7.4	347	738	1471	2586	4243	6651	9771
	7.6	335	714	1436	2499	4103	6408	9453
	7.8	321	696	1397	2404	3992	6243	9177
	8.0	308	679	1362	2349	3891	6056	8917
2.4	5.6	448	996	1989	3504	5748	9254	13653
	5.8	428	943	1885	3365	5435	8724	12952
	6.0	407	903	1792	3206	5201	8259	12328
	6.2	389	873	1737	3087	5003	8036	11853
	6.4	374	842	1673	2972	4812	7491	11332
	6.6	360	805	1623	2890	4661	7298	10866
	6.8	348	779	1559	2755	4461	6962	10462
	7.0	336	756	1519	2679	4320	6772	10254
	7.2	344	734	1468	2551	4198	6618	9800
	7.4	329	725	1431	2488	4089	6359	9427
	7.6	317	705	1389	2417	3989	6203	9185
	7.8	303	695	1363	2367	3871	6072	8895
	8.0	294	680	1329	2330	3791	5871	8673

TABLE 3.10 --Continued

SINGLE-SKIN TANK BARGE HULL WEIGHTS FOR CB = 0.80

B/T	L/B	BARGE LENGTH (FT)						
		250	325	400	475	550	625	700
2.6	5.6	427	952	1898	3343	5457	8660	12770
	5.8	406	916	1832	3208	5201	8250	12260
	6.0	389	871	1744	3070	4966	7897	11704
	6.2	373	842	1684	2951	4785	7638	11262
	6.4	359	809	1629	2863	4648	7250	10936
	6.6	346	784	1575	2770	4477	7096	10612
	6.8	328	771	1516	2667	4384	6809	10149
	7.0	316	745	1479	2596	4259	6597	9861
	7.2	326	745	1450	2500	4100	6405	9485
	7.4	315	720	1401	2466	3991	6282	9221
	7.6	305	700	1376	2415	3902	6088	9002
	7.8	292	691	1350	2367	3835	5915	8735
	8.0	283	680	1305	2303	3738	5780	8523
2.8	5.6	408	920	1829	3224	5263	8345	12337
	5.8	391	880	1776	3082	5069	7941	11712
	6.0	375	849	1697	2964	4843	7687	11444
	6.2	359	844	1644	2879	4678	7415	10952
	6.4	340	803	1588	2792	4541	7089	10610
	6.6	326	796	1555	2682	4398	6863	10198
	6.8	314	773	1506	2643	4266	6731	10035
	7.0	303	749	1472	2562	4154	6514	9689
	7.2	316	735	1438	2529	4077	6321	9296
	7.4	306	720	1404	2461	3957	6166	9078
	7.6	296	713	1365	2382	3827	6003	8900
	7.8	284	698	1341	2339	3766	5867	8660
	8.0	280	678	1321	2294	3706	5758	8454

TABLE 3.10 --Continued

SINGLE-SKIN TANK BARGE HULL WEIGHTS FOR CB = 0.80

B/T	L/B	BARGE LENGTH (FT)						
		250	325	400	475	550	625	700
	5.6	395	896	1785	3163	5088	8070	11822
	5.8	378	902	1711	2998	4957	7803	11497
	6.0	355	867	1670	2917	4760	7520	11096
	6.2	340	837	1631	2829	4616	7240	10745
	6.4	327	807	1582	2757	4475	6974	10370
	6.6	315	776	1547	2664	4339	6795	10080
3.0	6.8	302	774	1494	2626	4226	6591	9767
	7.0	293	765	1467	2555	4116	6434	9503
	7.2	311	747	1433	2501	4035	6275	9281
	7.4	304	724	1399	2436	3953	6142	9067
	7.6	299	714	1376	2396	3840	5979	8796
	7.8	285	698	1342	2358	3793	5895	8611
	8.0	276	689	1331	2313	3700	5740	8468

TABLE 3.11

SINGLE-SKIN TANK BARGE HULL WEIGHTS FOR CB = 0.85

B/T	L/B	BARGE LENGTH (FT)						
		250	325	400	475	550	625	700
2.2	5.6	475	1073	2141	3874	6382	10275	15128
	5.8	453	1015	2051	3610	5964	9682	14433
	6.0	434	964	1930	3472	5730	9124	13569
	6.2	414	926	1832	3325	5398	8716	12924
	6.4	398	881	1754	3164	5165	8030	12343
	6.6	382	859	1702	3054	5003	7821	11858
	6.8	366	826	1656	2958	4781	7411	11355
	7.0	353	797	1585	2847	4608	7168	10895
	7.2	361	769	1532	2692	4463	6973	10232
	7.4	348	748	1506	2629	4329	6777	10013
	7.6	336	726	1450	2535	4199	6583	9810
	7.8	323	715	1419	2457	4087	6373	9370
	8.0	308	704	1374	2403	3980	6259	9255
2.4	5.6	449	1001	2001	3584	5880	9382	14033
	5.8	429	950	1914	3389	5578	8946	13266
	6.0	408	917	1826	3262	5294	8495	12620
	6.2	390	881	1765	3144	5132	8052	12059
	6.4	376	849	1692	3017	4930	7749	11636
	6.6	361	822	1641	2905	4766	7412	11037
	6.8	348	793	1579	2817	4568	7171	10695
	7.0	336	783	1542	2712	4445	6941	10416
	7.2	345	761	1494	2613	4321	6740	9887
	7.4	330	744	1482	2548	4198	6501	9644
	7.6	317	729	1421	2501	4092	6364	9477
	7.8	304	711	1388	2452	3978	6204	9144
	8.0	294	692	1357	2388	3878	6046	8917

TABLE 3.11 --Continued

SINGLE-SKIN TANK BARGE HULL WEIGHTS FOR CB = 0.85

B/T	L/B	BARGE LENGTH (FT)						
		250	325	400	475	550	625	700
2.6	5.6	427	962	1915	3402	5576	8843	13238
	5.8	407	918	1846	3259	5338	8509	12634
	6.0	390	889	1760	3128	5135	8114	11937
	6.2	374	855	1700	3018	4890	7801	11473
	6.4	360	841	1651	2930	4758	7425	11126
	6.6	347	812	1595	2829	4575	7234	10824
	6.8	329	795	1562	2723	4448	6909	10401
	7.0	316	772	1515	2660	4338	6752	10121
	7.2	327	753	1472	2607	4242	6565	9647
	7.4	316	730	1437	2530	4095	6410	9429
	7.6	306	730	1401	2458	4002	6243	9190
	7.8	293	707	1370	2398	3898	6080	9002
	8.0	283	695	1342	2382	3825	5938	8764
2.8	5.6	409	931	1870	3266	5365	8478	12607
	5.8	391	888	1808	3152	5147	8228	11991
	6.0	376	880	1730	3059	4975	7869	11670
	6.2	360	860	1671	2936	4818	7588	11189
	6.4	341	833	1627	2854	4662	7259	10835
	6.6	327	808	1582	2766	4512	7050	10436
	6.8	315	782	1540	2709	4372	6843	10081
	7.0	304	782	1500	2629	4276	6637	9903
	7.2	317	751	1466	2565	4157	6523	9496
	7.4	310	740	1429	2549	4079	6329	9296
	7.6	302	723	1396	2450	3956	6190	9088
	7.8	291	707	1377	2426	3868	6030	8875
	8.0	287	705	1356	2346	3780	5889	8734

TABLE 3.11 --Continued

SINGLE-SKIN TANK BARGE HULL WEIGHTS FOR CB = 0.85

B/T	L/B	BARGE LENGTH (FT)						
		250	325	400	475	550	625	700
	5.6	396	927	1826	3222	5235	8282	12258
	5.8	378	909	1761	3067	5055	7961	11698
	6.0	357	874	1706	2993	4871	7740	11378
	6.2	341	846	1652	2886	4752	7423	10923
	6.4	328	825	1619	2832	4590	7245	10622
	6.6	316	809	1571	2757	4457	6962	10326
3.0	6.8	304	797	1539	2673	4323	6779	9971
	7.0	304	774	1498	2616	4223	6627	9759
	7.2	314	756	1462	2600	4132	6431	9506
	7.4	309	751	1442	2517	4043	6284	9266
	7.6	300	732	1416	2461	3956	6148	9038
	7.8	286	713	1381	2408	3876	6003	8872
	8.0	277	698	1353	2399	3792	5870	8701

TABLE 3.12

DOUBLE BOTTOM TANK BARGE HULL WEIGHTS FOR CB=0.80

B/T	L/B	BARGE LENGTH (FT)						
		250	325	400	475	550	625	700
2.2	5.6	552	1189	2312	4039	6539	10430	15184
	5.8	526	1135	2198	3867	6184	9863	14382
	6.0	505	1091	2052	3678	5877	9295	13580
	6.2	483	1049	1980	3508	5555	8896	12839
	6.4	464	1009	1889	3362	5317	8260	12324
	6.6	447	970	1817	3211	5087	8001	11765
	6.8	432	932	1718	3104	4908	7678	11267
	7.0	414	903	1684	2977	4737	7291	10789
	7.2	401	890	1620	2819	4565	6998	10213
	7.4	390	858	1573	2729	4392	6779	9839
	7.6	378	828	1528	2641	4267	6576	9505
	7.8	368	804	1477	2568	4145	6339	9142
	8.0	357	786	1422	2477	4019	6149	8884
2.4	5.6	524	1138	2189	3781	6064	9577	14111
	5.8	501	1089	2084	3607	5779	9062	13303
	6.0	481	1045	1951	3437	5493	8721	12601
	6.2	462	1004	1864	3309	5237	8326	11964
	6.4	442	983	1823	3173	5051	7804	11487
	6.6	426	947	1745	3054	4833	7530	11037
	6.8	412	912	1664	2930	4690	7234	10583
	7.0	399	881	1615	2835	4512	6926	10197
	7.2	389	855	1547	2726	4369	6695	9696
	7.4	374	843	1499	2633	4211	6491	9413
	7.6	363	815	1462	2563	4102	6252	9077
	7.8	353	790	1427	2488	3963	6084	8800
	8.0	342	772	1390	2391	3858	5901	8547

TABLE 3.12 --Continued

DOUBLE BOTTOM TANK BARGE HULL WEIGHTS FOR CB=0.80

B/T	L/B	BARGE LENGTH (FT)						
		250	325	400	475	550	625	700
	5.6	504	1100	2058	3595	5744	9061	13143
	5.8	481	1066	1989	3433	5523	8614	12473
	6.0	460	1024	1893	3289	5275	8218	11820
	6.2	443	982	1817	3171	5032	7888	11354
	6.4	428	946	1757	3041	4825	7484	10898
	6.6	413	915	1680	2937	4663	7224	10519
2.6	6.8	397	903	1610	2854	4503	6980	10119
	7.0	385	870	1562	2764	4355	6683	9757
	7.2	375	840	1523	2650	4219	6438	9382
	7.4	363	816	1466	2561	4092	6284	9069
	7.6	351	796	1430	2499	3966	6108	8817
	7.8	340	786	1398	2447	3845	5936	8577
	8.0	331	762	1372	2350	3763	5808	8416
	5.6	484	1076	2005	3457	5532	8602	12508
	5.8	465	1036	1922	3297	5314	8247	11966
	6.0	447	994	1836	3179	5095	7908	11345
	6.2	431	976	1761	3082	4845	7613	10917
	6.4	414	938	1707	2974	4687	7262	10502
	6.6	399	904	1647	2860	4533	6993	10153
2.8	6.8	384	877	1577	2762	4380	6760	9771
	7.0	370	852	1530	2677	4236	6536	9477
	7.2	363	839	1488	2610	4091	6333	9230
	7.4	352	812	1460	2551	4030	6169	8990
	7.6	341	789	1432	2456	3920	6011	8755
	7.8	332	771	1390	2437	3817	5909	8567
	8.0	333	750	1375	2352	3753	5778	8347

TABLE 3.12 --Continued

DOUBLE BOTTOM TANK BARGE HULL WEIGHTS FOR CB=0.80

B/T	L/B	BARGE LENGTH (FT)						
		250	325	400	475	550	625	700
	5.6	472	1066	1932	3340	5382	8325	12066
	5.8	454	1030	1864	3229	5161	7980	11548
	6.0	434	985	1797	3110	4962	7702	10970
	6.2	416	952	1736	3000	4723	7393	10587
	6.4	401	923	1676	2897	4590	7086	10186
	6.6	386	907	1628	2815	4406	6871	9963
3.0	6.8	373	874	1570	2740	4308	6650	9619
	7.0	360	847	1531	2647	4182	6439	9413
	7.2	355	824	1486	2591	4090	6308	9192
	7.4	355	799	1467	2523	4000	6195	8960
	7.6	343	780	1427	2487	4930	6053	8794
	7.8	332	763	1420	2431	3817	5899	8594
	8.0	323	742	1380	2388	3761	5789	8462

TABLE 3.13

REGRESSION EQUATIONS FOR BARGE LENGTH, DEADWEIGHT, AND HULL STEEL WEIGHT

Regression Type	Barge Type	C_B	REGRESSION EQUATION
Barge Hull Steel Weight f(L, L/B, B/T)	Single-Skin	0.75	$e^{-9.921}\,(L/B)^{-1.091}\,(B/T)^{-0.372}\,(L)^{3.290}$
	"	0.80	$e^{-10.063}\,(L/B)^{-1.076}\,(B/T)^{-0.350}\,(L)^{3.308}$
	"	0.85	$e^{-10.188}\,(L/B)^{-1.057}\,(B/T)^{-0.331}\,(L)^{3.323}$
	Double-Bottom	0.80	$e^{-8.784}\,(L/B)^{-1.175}\,(B/T)^{-0.389}\,(L)^{3.147}$
Barge Hull Steel Weight f(L/B, B/T, DWT)	Single-Skin	0.75	$e^{-6.206}\,(L/B)^{1.348}\,(B/T)^{0.884}\,(DWT)^{1.104}$
	"	0.80	$e^{-6.396}\,(L/B)^{1.368}\,(B/T)^{0.909}\,(DWT)^{1.111}$
	"	0.85	$e^{-6.569}\,(L/B)^{1.391}\,(B/T)^{0.930}\,(DWT)^{1.117}$
	Double-Bottom	0.80	$e^{-5.150}\,(L/B)^{1.126}\,(B/T)^{0.800}\,(DWT)^{1.049}$
Barge Length f(L/B, B/T, DWT)	Single-Skin	0.75	$e^{1.128}\,(L/B)^{0.742}\,(B/T)^{0.382}\,(DWT)^{0.336}$
	"	0.80	$e^{1.108}\,(L/B)^{0.739}\,(B/T)^{0.381}\,(DWT)^{0.336}$
	"	0.85	$e^{1.088}\,(L/B)^{0.737}\,(B/T)^{0.379}\,(DWT)^{0.336}$
	Double-Bottom	0.80	$e^{1.154}\,(L/B)^{0.731}\,(B/T)^{0.378}\,(DWT)^{0.333}$
Barge Deadweight f(L/B, B/T, L)	Single-Skin	0.75	$e^{-3.359}\,(L/B)^{-2.209}\,(B/T)^{-1.137}\,(L)^{2.978}$
	"	0.80	$e^{-3.295}\,(L/B)^{-2.199}\,(B/T)^{-1.133}\,(L)^{2.976}$
	"	0.85	$e^{-3.237}\,(L/B)^{-2.191}\,(B/T)^{-1.129}\,(L)^{2.975}$

NOTES: Barge depth is assumed to be the minimum allowed by the Freeboard rules presented in Table 2.5. DWT refers to the barge cargo deadweight.

TABLE 3.14

COMPARISONS OF BARGE DESIGN MODEL'S HULL STEEL WEIGHT ESTIMATES
WITH ACTUAL BARGE HULL STEEL WEIGHTS

Barge Particulars L X B X D	Draft	DWT	Remarks	Assumed Scanting Length	Assumed Block Coefficient	Actual Hull Steel Weight	Calculated Hull Steel Weight	% Error
546'0"x85'0"x38'9"	32'0"	31,500	Single-Skin 2nd Generation Tank Barge	546'0"	0.86	4180 LT	4558 LT	+ 9%
				520'0"	0.90	"	4154	- 1%
581'0"x95'0"x46'0"	36'0"	40,500	Single-Skin Rigid Linked 3rd Generation Tank Barge	581'0"	0.83	5398 LT	6123 LT	+13%
				525'0"	0.90	"	5156	- 4%
450'0"x80'0"x32'6"	26'3"	20,000	Single-Skin 2nd Generation Tank Barge	450'0"	0.85	2181 LT	2668 LT	+22%
				420'0"	0.90	"	2330	+ 6%
269'3"x54'0"x17'6"	13'8"	4,200	Single-Skin 1st Generation Tank Barge	269'3"	0.85	588 LT	521 LT	-11%
				250'0"	0.90	"	462	-21%

CHAPTER 4

TUG-BARGE POWERING PROGRAM TO ESTIMATE DHP

4.1 Introduction

The tug-barge powering program "power" has been developed to provide an estimate of the horsepower required to be delivered to the propeller to drive an integrated tug-barge unit of given size and form at a specified speed. This delivered horsepower (DHP) estimate is used by the drop-and-swap program to estimate the horsepower of the machinery to be installed onboard the tug (IHP) and the horsepower required to be delivered to the shaft (SHP). The IHP estimate is used in a regression equation formulated by Sharp (1975) to estimate cost of propulsion machinery. The SHP estimate is used to determine fuel costs. This is done by simply multiplying SHP by cost per unit fuel, by voyage duration, and by specific fuel consumption. Thus, a reasonably accurate estimate of DHP is necessary to ensure that tug capital and operating costs used by the drop-and-swap program are reasonable.

The barge powering program is basically a table lookup and interpolation/extrapolation routine that utilizes the

tankship resistance and propulsion data of Tsuchida (1969) which is based on hull lines shown in Figure 4.1. From the resistance data the program determines the effective horsepower (EHP) needed to propel the tug-barge unit at the specified speed. Then the EHP, self-propulsion factors, and some assumed propeller particulars are fed into the Triantafyllou (1977) preliminary propeller design program to obtain estimates of the propeller's propulsive efficiency and consequently the horsepower required to be delivered to the propeller.

The tankship data of Tsuchida (1969) is used since, unfortunately, no series tests of large ocean going barge forms have been conducted. However, the tankship model tests use forms similar to the full-bodied integrated tug-barge hulls currently in operation today. Thus, no significant error should result from using this data for tug-barge data.

The propeller design program of Triantafyllou (1977) is used since this program uses a mathematical formulation of Wageningen B-series that is corrected for laminar flow effects. It provides very good agreement with model tests of commercial type hulls and propellers and thus is appropriate for use in the barge powering estimate. Since this program is documented in Triantafyllou (1977), the details of its logic will not be presented here.

In the text that follows, the logical structure of the powering program is presented. This is followed by the pre-

sentation and discussion of the table data and sample program outputs.

4.2 Logical Structure

The logical structure of the program is shown in flow chart form in Figure 4.2. In the text following, each block of the flow chart is explained (except for the propeller efficiency program explained in Triantafyllou (1977)) so that the tables and assumptions used are enumerated.

4.2.1 Input of Parameter Values

The barge powering program requires as input the tug-barge unit's principal dimensions, block coefficient, and speed. Specifically, the following values must be provided for the tug-barge unit design:

"l": length at the waterline

"b": molded breadth

"dr": summer waterline draft

"cb": barge block coefficient

"v": speed

Note, except for "l" which corresponds to the length of the tug integrated with the barge, barge values are used above. This is because the tug is normally designed to integrate smoothly with the barge's lines so that the combined unit's dimensions and block coefficient (except for "l") are not significantly different from the barge alone.

These values are immediately converted to ratios--length breadth ("lb"="l"/"b"), breadth draft ("bt"="b"/"dr"), and Froude number ("fn"=0.2976x"v"/sqrt("l"))--which are used as inputs into the subroutines "resist" and "propfactors" described later. Limitations on the ranges of values that these ratios and the block coefficient may take are as follows:

$$0.75 \leq \text{"cb"} \leq 0.85$$

$$6.0 \leq \text{"lb"} \leq 8.0$$

$$\text{"fn"} \leq 0.22$$

$$2.0 \leq \text{"bt"} \leq 3.25$$

Either of the first two and both of the last two limitations must hold for the program to operate without significant error. The rational of these limitations will be explained in the following text. Following the calculation of these ratios the program calls the subroutine "propfactors".

4.2.2 Calculation of Self-Propulsion Factors

The subroutine "propfactors" is a simple linear interpolation and quadratic extrapolation routine that provides values for wake fraction (W_T="wa"), thrust deduction fraction (t="th"), and relative rotative efficiency (η_R ="hr") as a function of "lb" and "cb". These self-propulsion factor values are interpolated or extrapolated from the tables of (1-t), (1-W_T), and (η_R) shown in Table 4.1 which were derived

from graphs presented in Tsuchida (1969, Figs. 60 and 63). These tables provide a range of "lb" values from 6.2 to 7.6 by 0.1 increments and a range of "cb" values from 0.770 to 0.840 by 0.005 increments for a Froude number of 0.16.

For "lb" and "cb" values that fall within the table ranges, the subroutine "propfactors" obtains the self-propulsion factors through a simple two dimensional linear interpolation of table values. If either "lb" or "cb" fall outside of the table ranges, then the subroutine utilizes quadratic extrapolation to extend the two nearest table entries for the variable that exceeds the table range and then uses linear interpolation between these two values for the other variable. For example, if "cb" is 0.75 and "lb" is 7.15 then the subroutine will use quadratic extrapolation to find the table extrapolation of "cb" to 0.75 for "lb" of 7.1 and 7.2. Then the subroutine will linearly interpolate between these two extrapolations to obtain the value for "lb" of 7.15.

To ensure that the extrapolation will not yield gross errors, table extensions have been limited to:

$$0.75 \leq \text{"cb"} \leq 0.85$$

or

$$6.0 \leq \text{"lb"} \leq 8.0$$

If either of these ranges are exceeded or if the values of both "lb" and "cb" exceed table ranges, the program indicates this error has occurred ("iprop"=1) to the calling program "power" and does not calculate the self-propulsion factors.

After obtaining the self-propulsion factors for the specified value of "lb" and "cb", the program corrects the wake fraction for Froude numbers greater than 0.16. The other factors do not need such correction according to Tsuchida (1969, pp. 7-8). Following this, the subprogram "power" calls the subroutine "resist" to obtain the residual resistance coefficient.

4.2.3 Calculation of Residual Resistance Coefficient

The subroutine "resist" is a four dimensional linear and quadratic extrapolation/interpolation routine that provides values for the residual resistance coefficient ("cr") as a function of "lb", "cb", "fn", and "bt". These residual resistance coefficients are interpolated and extrapolated from tables shown in Table 4.2 which were derived from the graphs presented in Tsuchida (1969, Figs. 7-32). These tables provide a range of "cb" values from 0.775 to 0.835 by 0.01 increments, a range of "lb" values from 6.2 to 7.6 by 0.2 increments, and a range of "fn" values from 0.14 to 0.22 by 0.01 increments (except between 0.14 and 0.16) for "bt" values of 2.46 and 2.76. These data are rerepresented in graphic form with "cr" as a function of "fn" for a range of "lb" values, with separate graphs for different "cb" and "bt" values in Figure 4.3.

After the subroutine reconstructs the residual resistance coefficient array by creating linearly interpolated

entries for a Froude number of 0.15, it proceeds with quadratic interpolation or extrapolation with respect to Froude number. That is, for the four "cb" and "lb" table values that bracket the inputted "cb" and "lb" value (for each "bt" value), the subroutine fits a parabola through the three closest Froude number table entries and then quadratically interpolates or extrapolates for the inputted Froude number. It then does a two dimensional linear interpolation among these four interpolated/extrapolated values with respect to "cb" and "lb". After this is done for both "bt" tables, the subroutine finishes by linearly interpolating or extrapolating between the two "bt" values for the inputted "bt" value.

To insure that the quadratic extrapolation routine does not yield gross errors, Froude number inputs are limited to a value of 0.22. No lower bound is specified since the values are almost constant for low Froude numbers so that extrapolation will cause little error.

After obtaining the residual resistance coefficient for the specified values of "lb", "bt", "cb", and "fn", the subprogram "power" proceeds to calculate the inputs for the subroutine "prop".

4.2.4 Calculation of Inputs for Subroutine "prop"

The propeller efficiency subroutine "prop" requires as input the self-propulsion factors previously calculated as well as some particulars concerning the propeller, i.e., its

diameter, number of blades, and expanded area ratio. It also requires an estimate of effective horsepower needed to drive the tug-barge hull at the specified speed (EHP) through still water.

The latter value is proportional to the unit's speed and hull resistance. The hull resistance is, in turn, proportional to the vessel's wetted surface area, square of its speed, and its total resistance coefficient. The wetted surface area is estimated by the formula given in Tsuchida (1969, p.11):

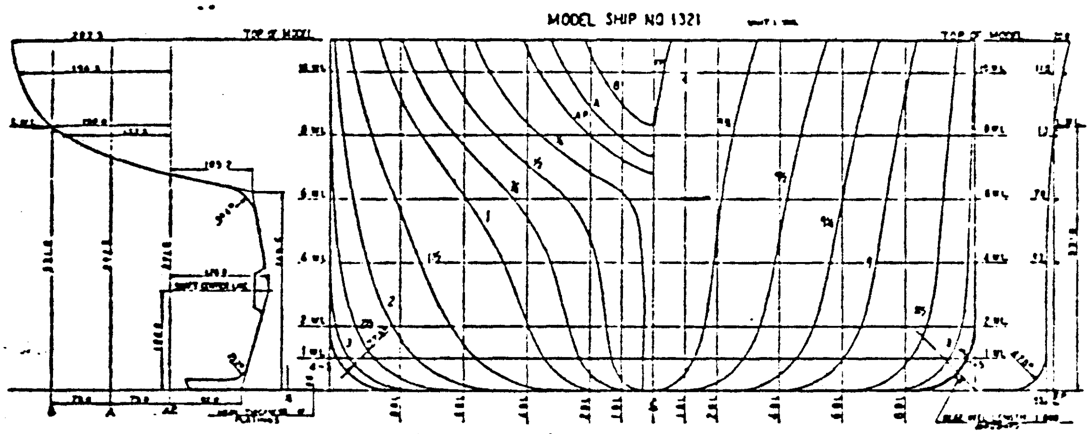
"wsa" = 1.81 x "l" x "dr" + "cb x "l" x "b"

And, the total resistance coefficient is determined by summing the residual resistance coefficient (obtained in the previous section), the frictional resistance coefficient (calculated by the ATTC formula via an iterative routine), and assumed roughness allowance of 0.0004.

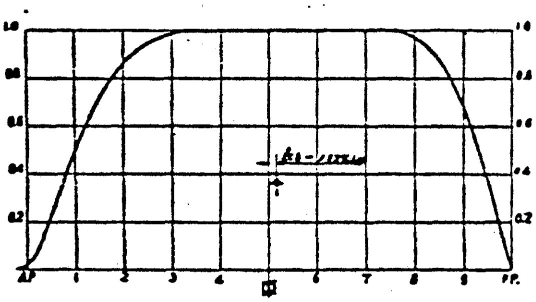
As for the propeller particulars, these have been assumed to agree with the propeller used in the tanker series given in Tsuchida (see Fig. 4.1). Specifically, the propeller is assumed to have four blades, a diameter equal to 0.636 of the barge draft, and an expanded area ratio estimated by a semi-empirical formula recommended by Wageningen for his B-screw series.

Following the determination of the above inputs, subroutine "prop" is called. As stated previously, this subroutine determines via a mathematical formulation of the

propeller series curves the propulsive efficiency of the propeller (η) and the horsepower required to be delivered to propeller ($DHP = EHP/\eta$). The subprogram "power" then terminates upon returning the value of "dhp" to the main program "drop-and-swap".

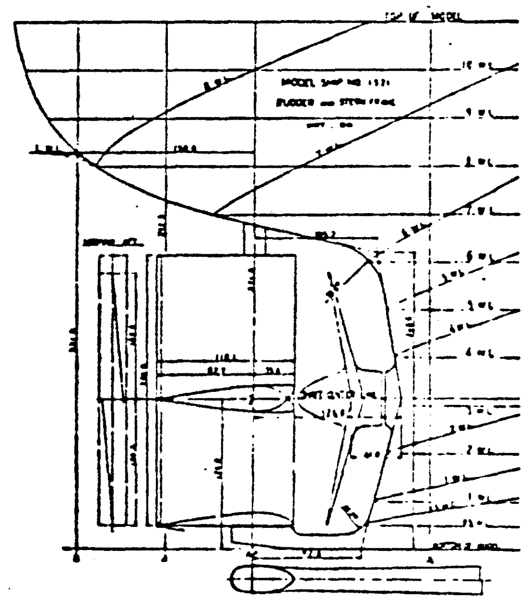


Body Plan, Stem & Stern Contour [M.S. No. 1321]



Prismatic Curve

Prismatic Curve of
M.S. No. 1321



Stern Frame & Rudder
of M.S. No. 1321

SOURCE: Tsuchida (1969)

FIGURE 4.1

TANK-SHIP HULL LINES USED IN OBTAINING RESIDUAL RESISTANCE
COEFFICIENT AND SELF-PROPULSION FACTOR DATA

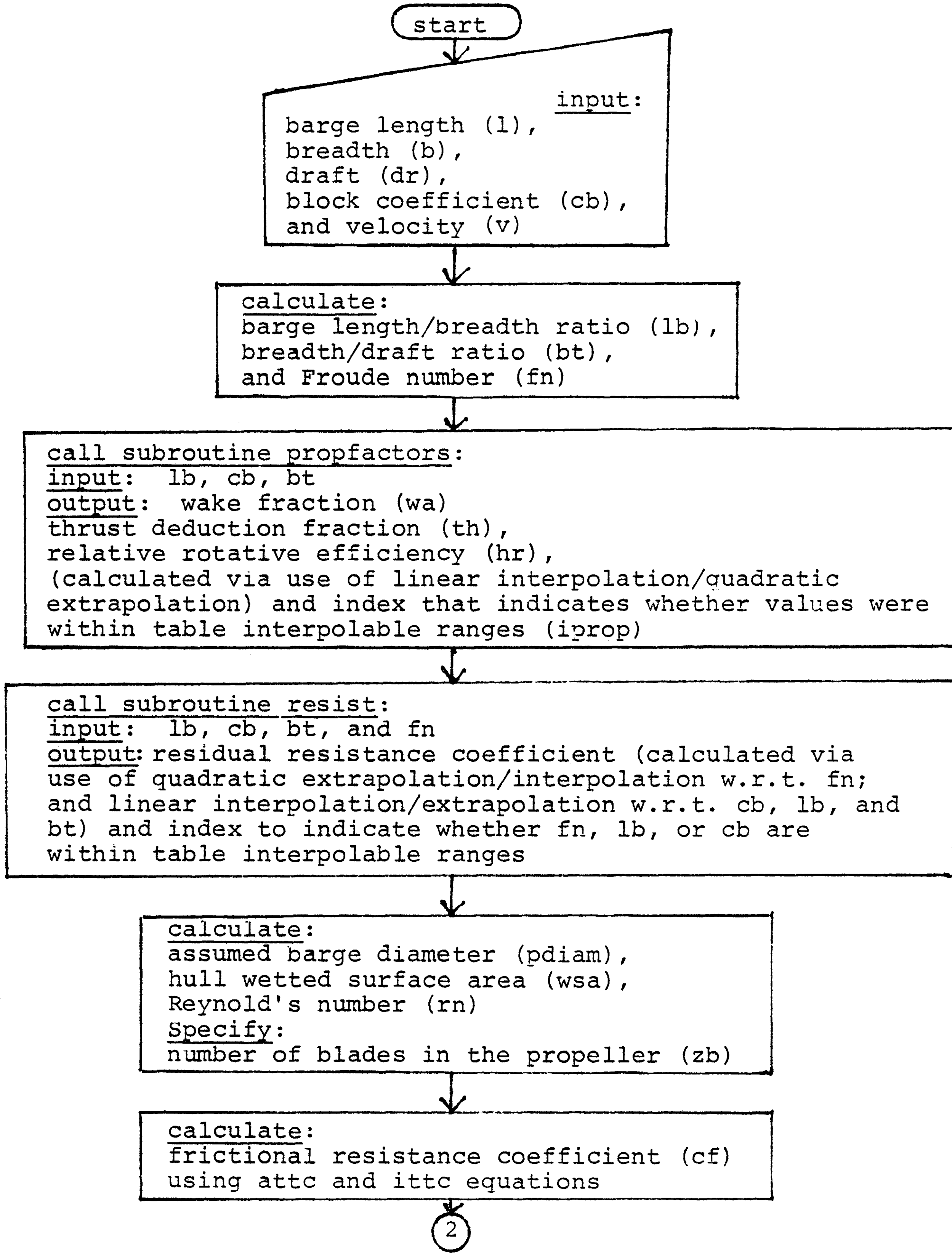


FIGURE 4.2

SUMMARY FLOW CHART FOR POWERING SUBPROGRAM

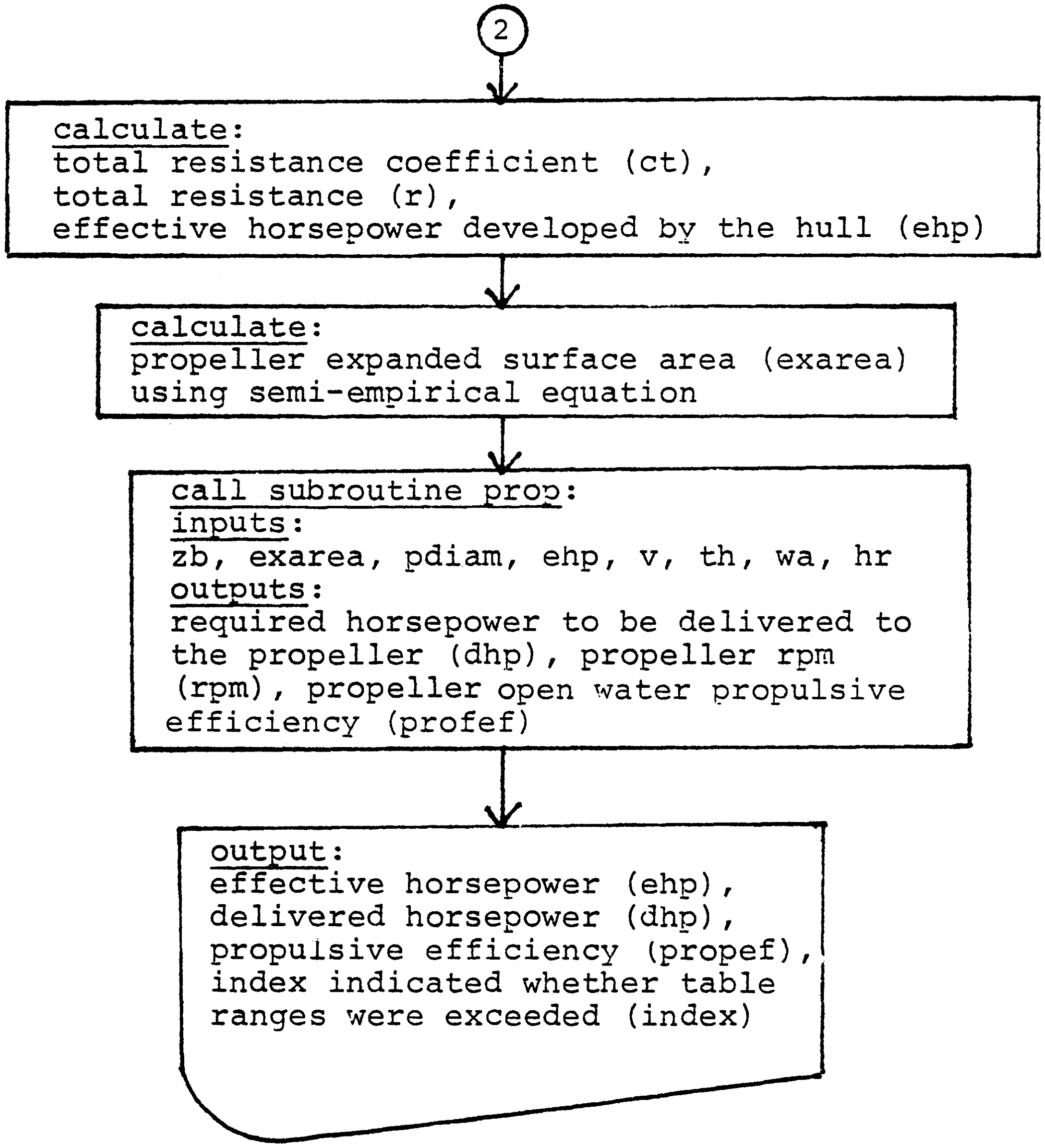


FIGURE 4.2...(Continued)

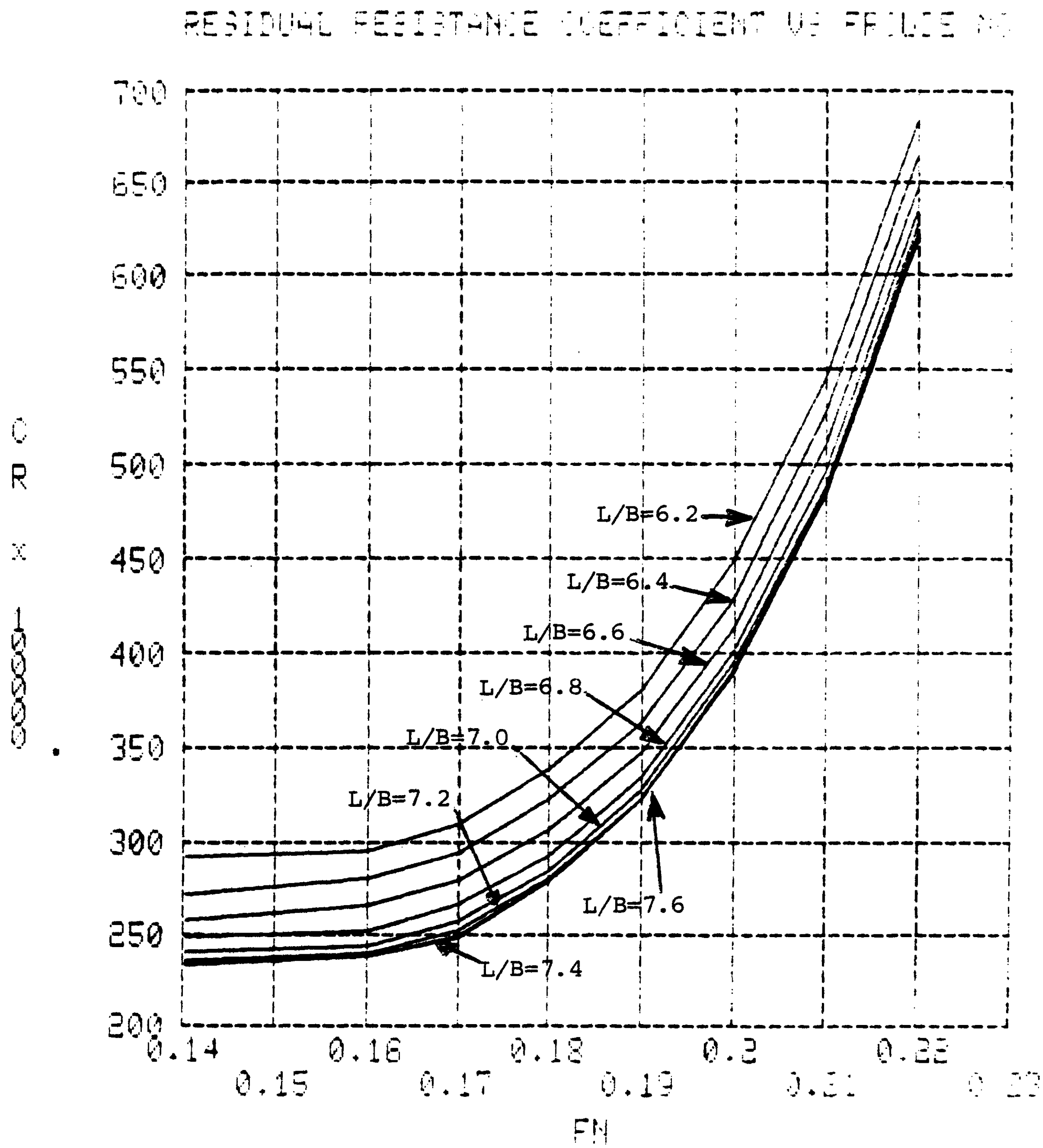


FIGURE 4.3

PLOT OF C_R VS. FN FOR C_B=0.775 AND B/T=2.46

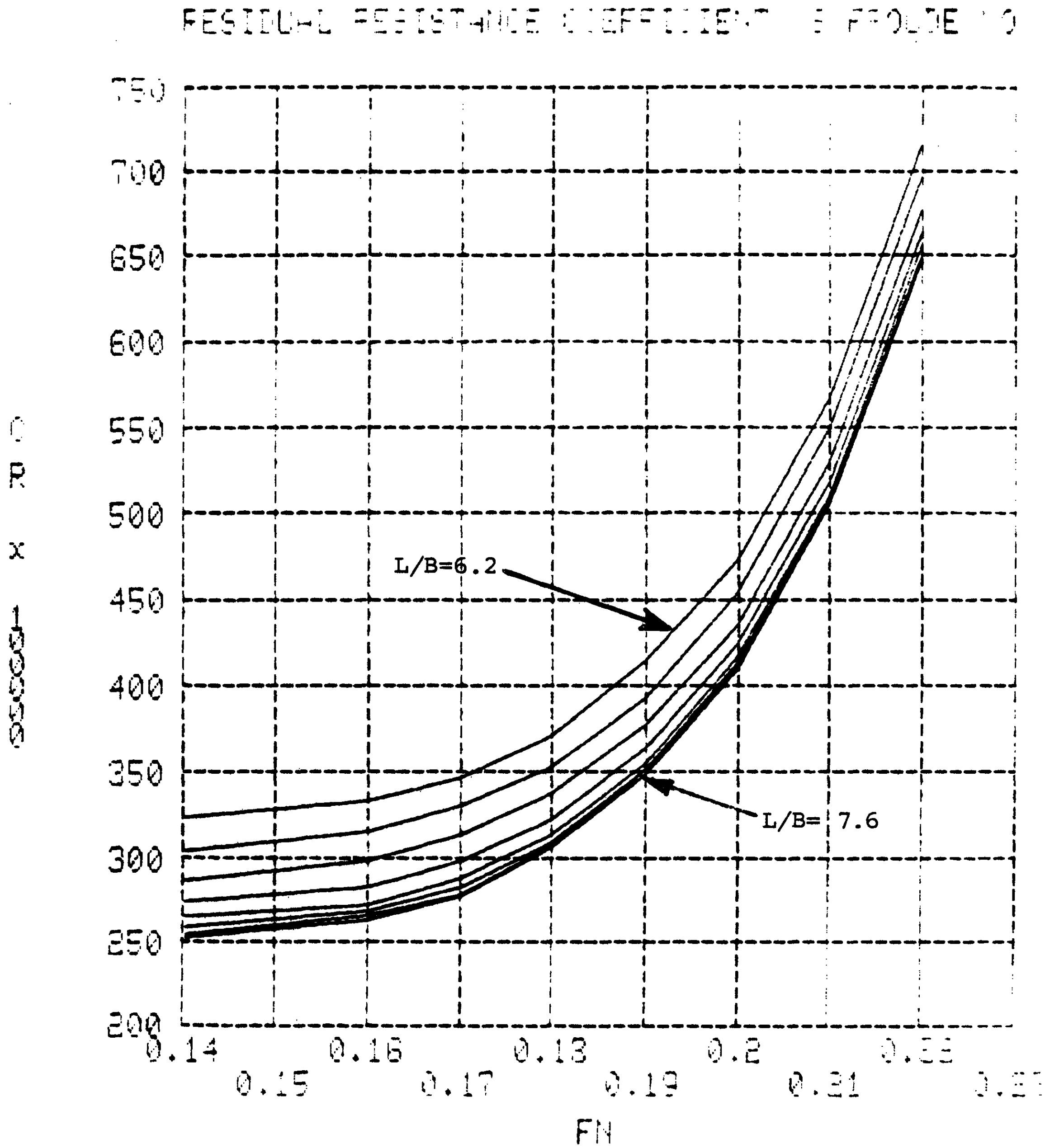


FIGURE 4.3

PLOT OF C_R VS. FN FOR C_B=0.785 AND B/T=2.46

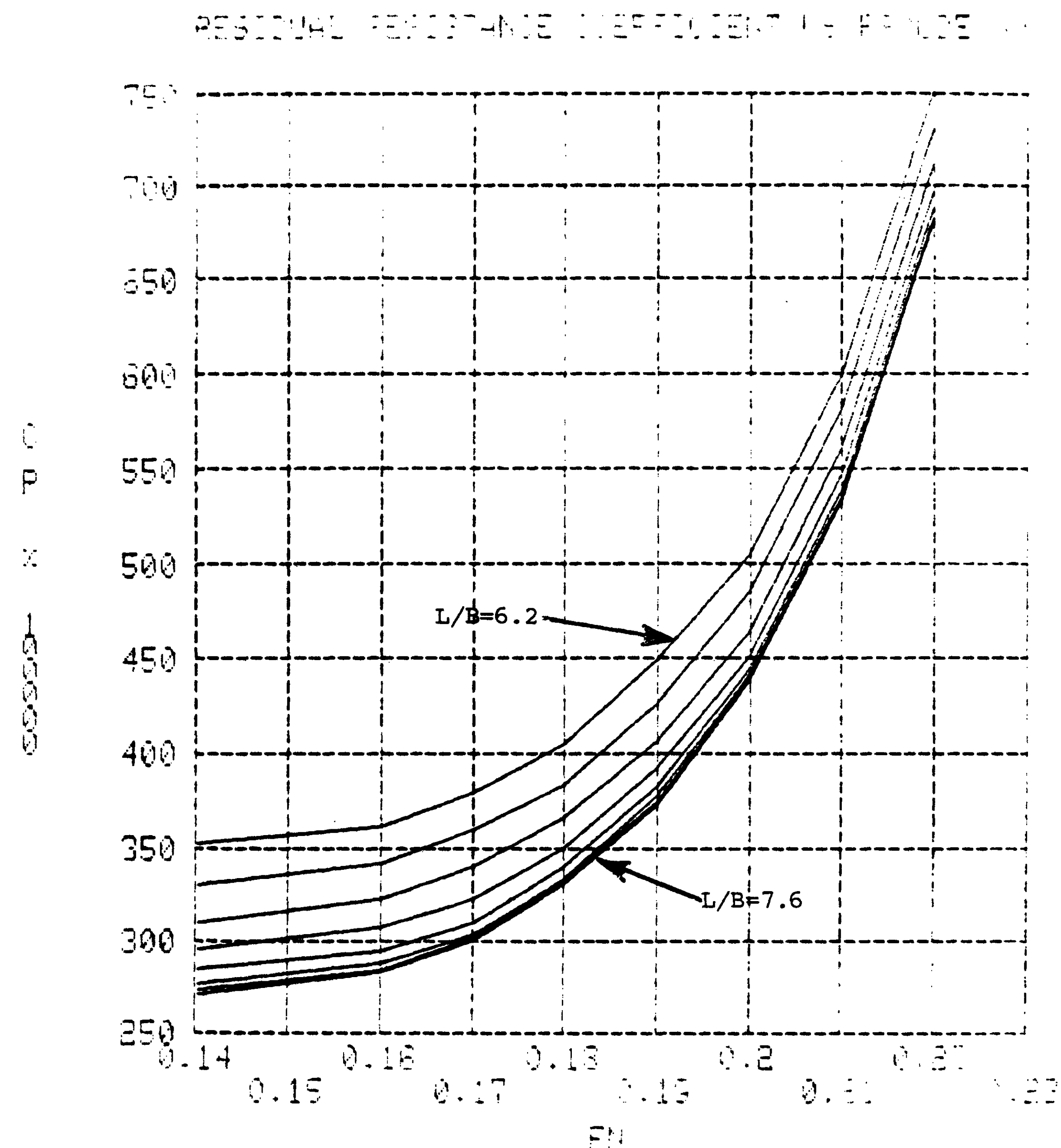


FIGURE 4.3

PLOT OF C_R VS. FN FOR C_B=0.795 AND B/T=2.46

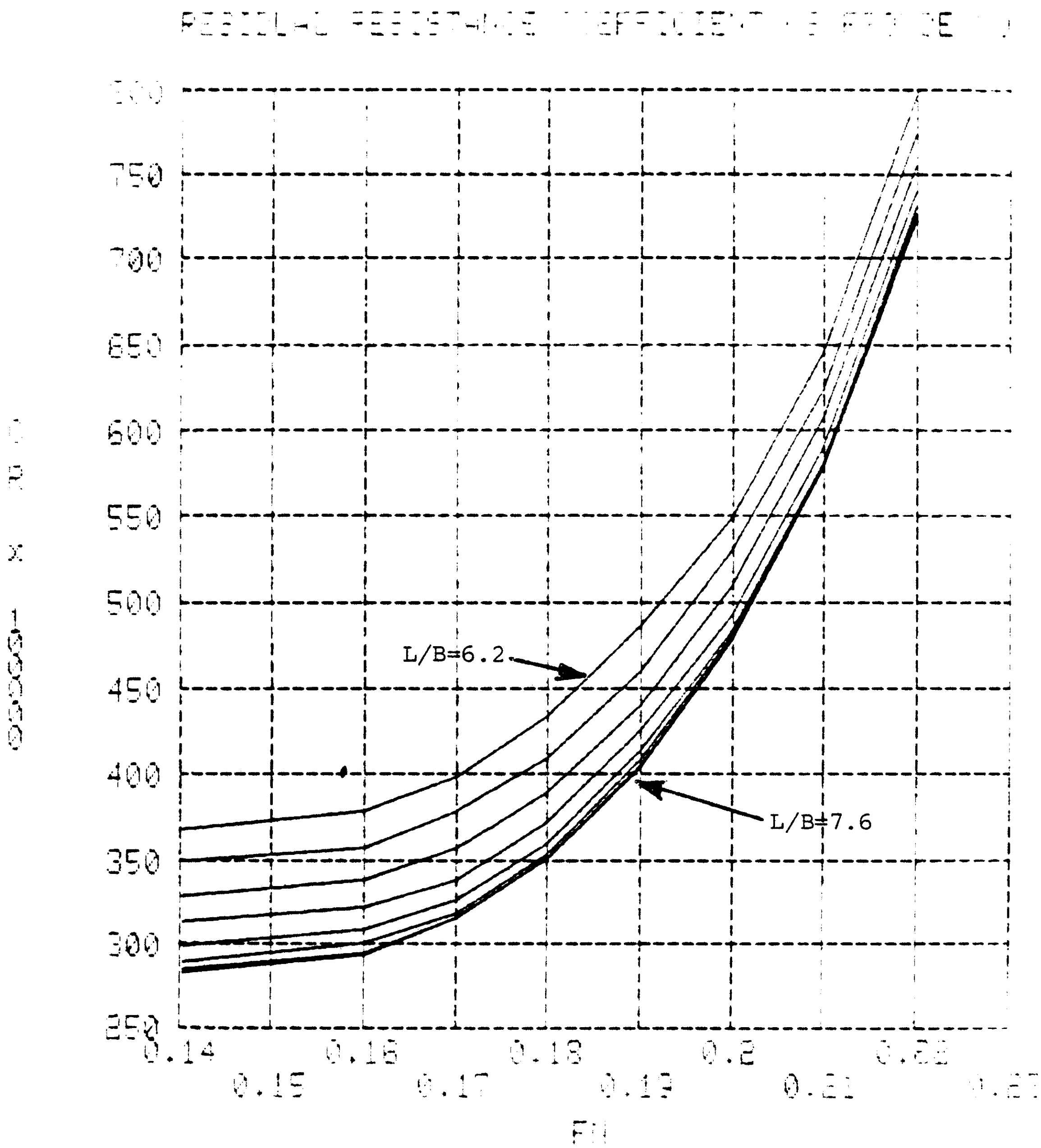


FIGURE 4.3

PLOT OF C_R VS. FN FOR C_B=0.805 AND B/T=2.46

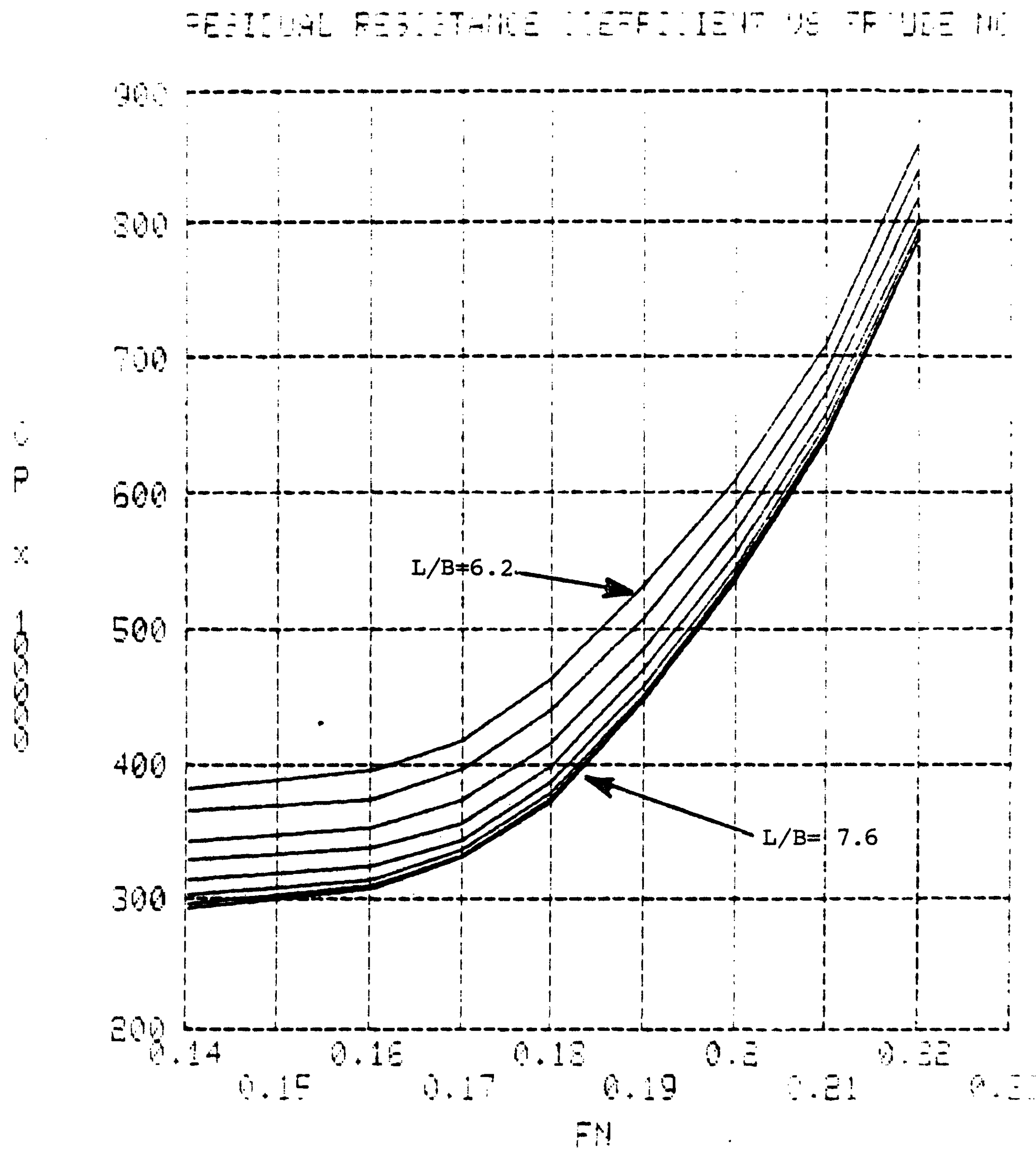


FIGURE 4.3

PLOT OF C_R VS. FN FOR C_B=0.815 AND B/T=2.46

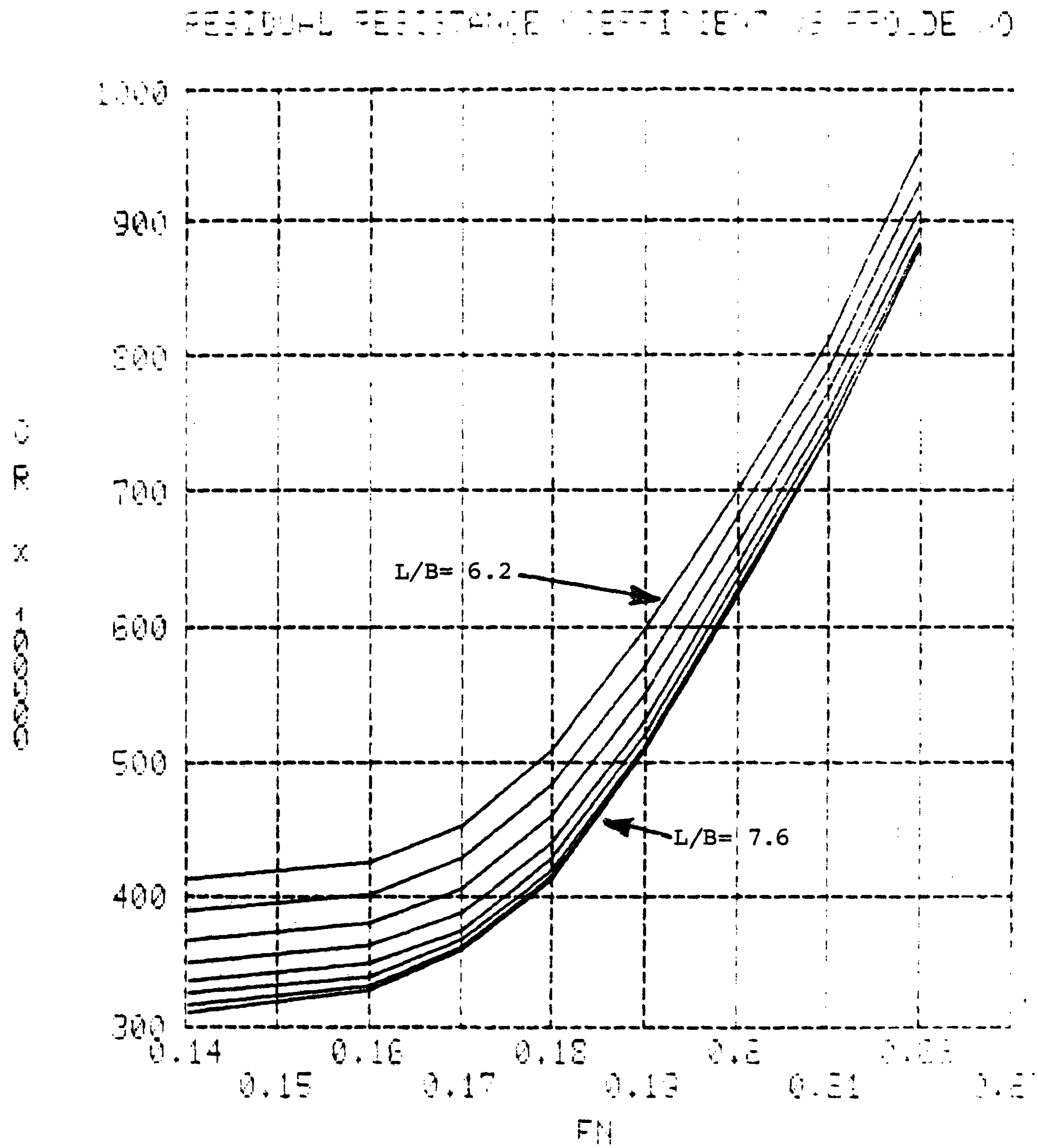


FIGURE 4.3

PLOT OF C_R VS. FN FOR C_B=0.825 AND B/T=2.46

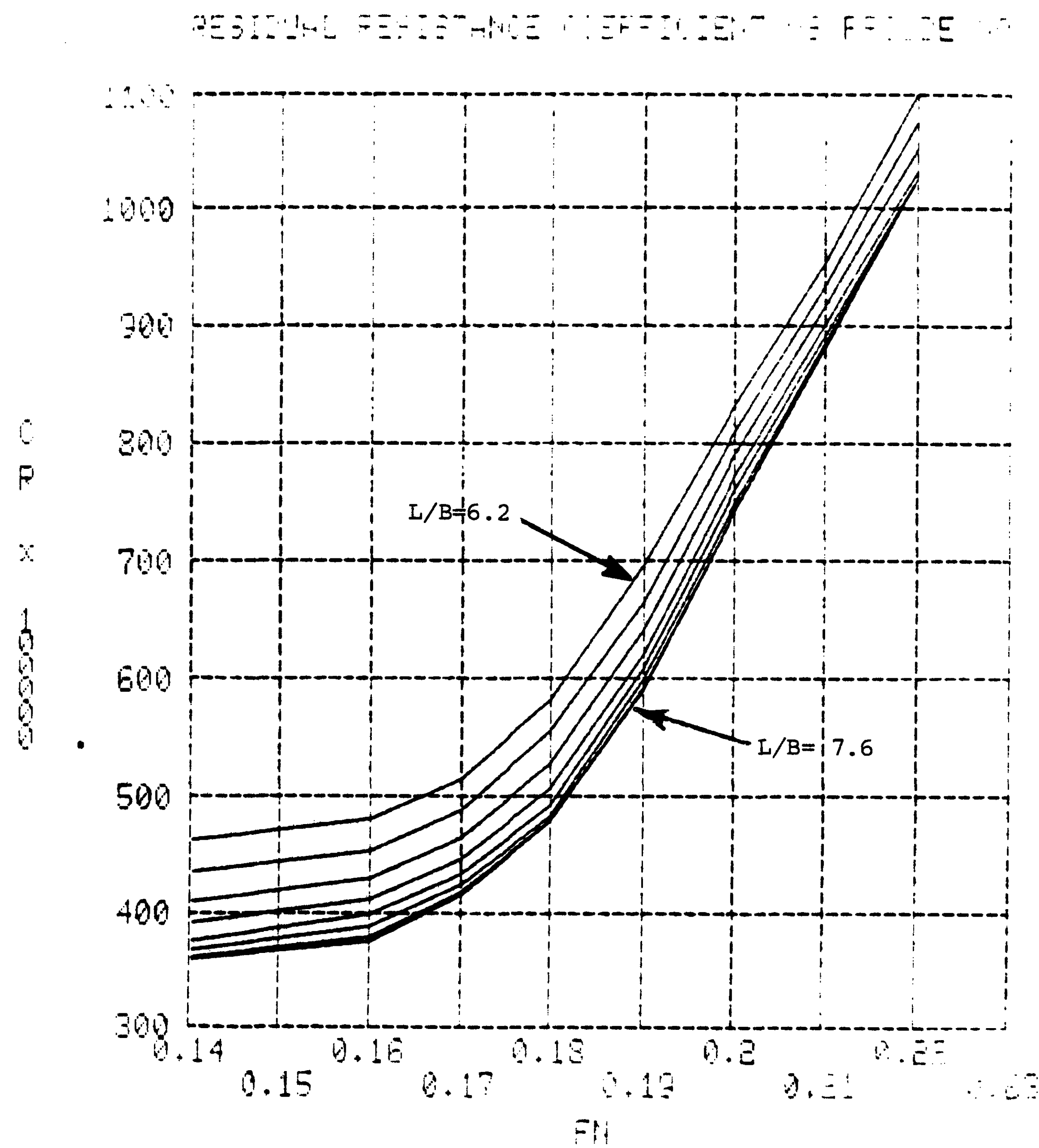


FIGURE 4.3

PLOT OF C_R VS. FN FOR C_B=0.835 AND B/T=2.46

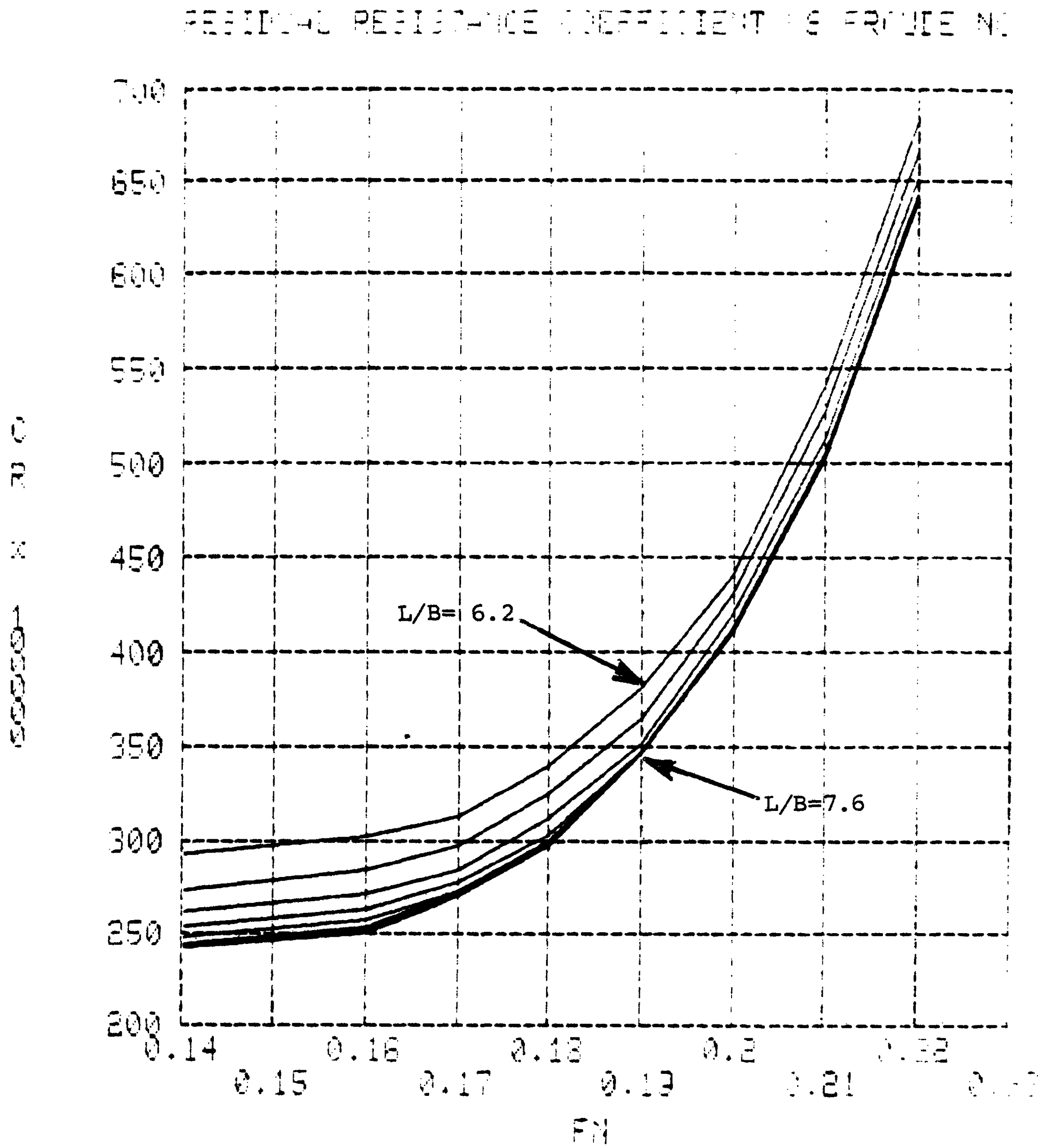


FIGURE 4.3

PLOT OF C_R VS. FN FOR C_B=0.775 AND B/T=2.76

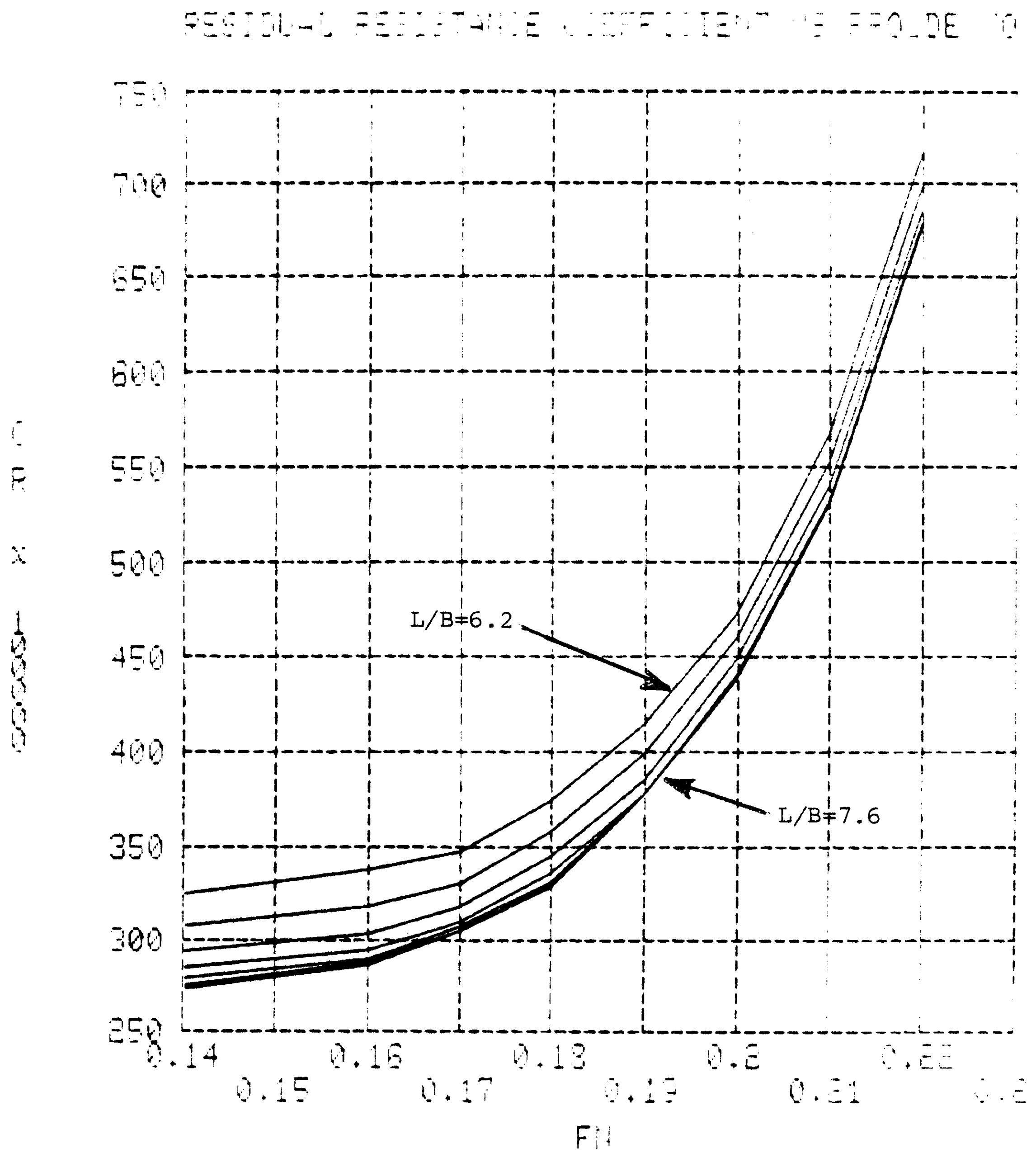


FIGURE 4.3

PLOT OF C_R VS. FN FOR C_B=0.785 AND B/T=2.76

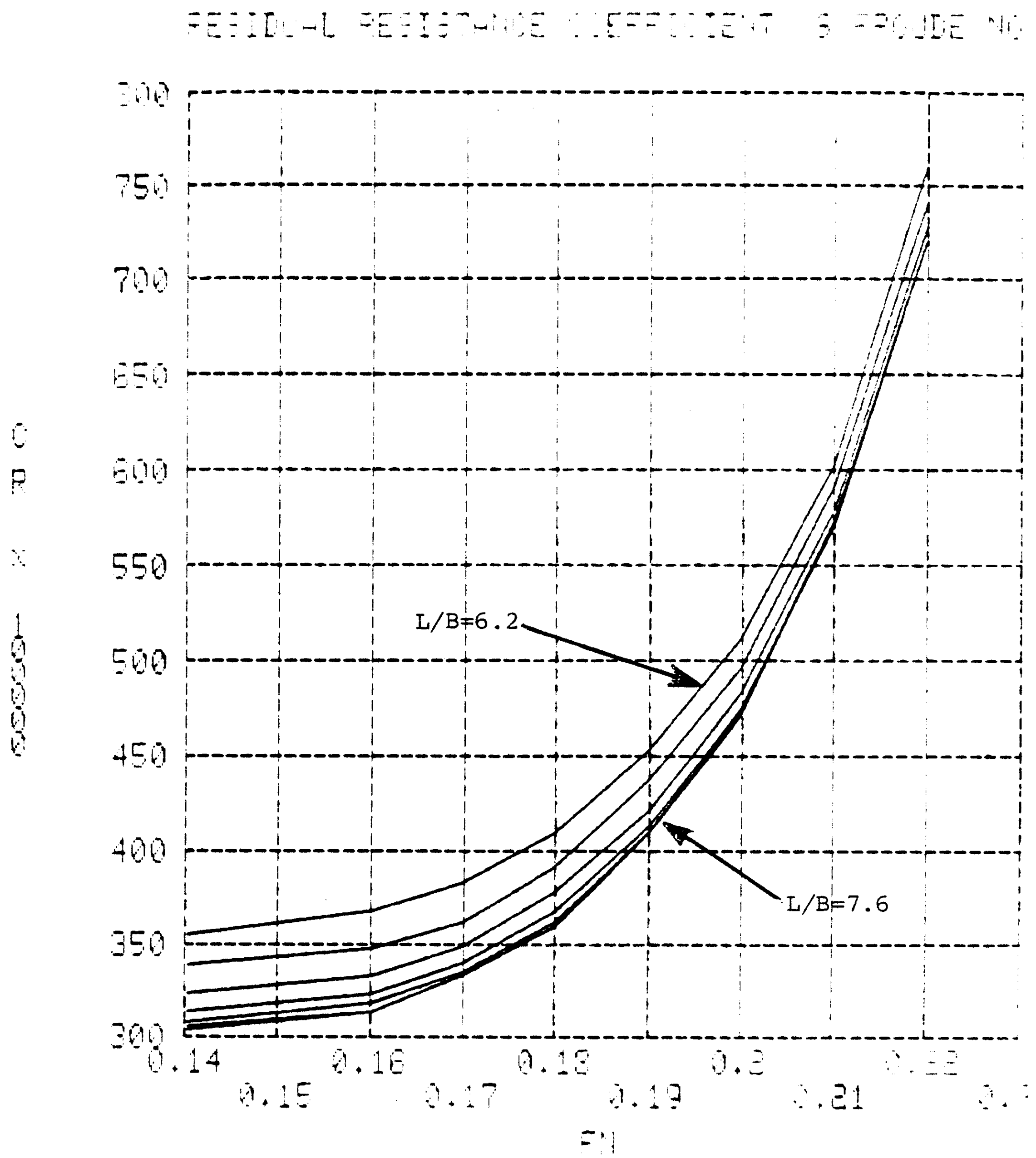


FIGURE 4.3

PLOT OF C_R VS. FN FOR C_B=0.795 AND B/T=2.76

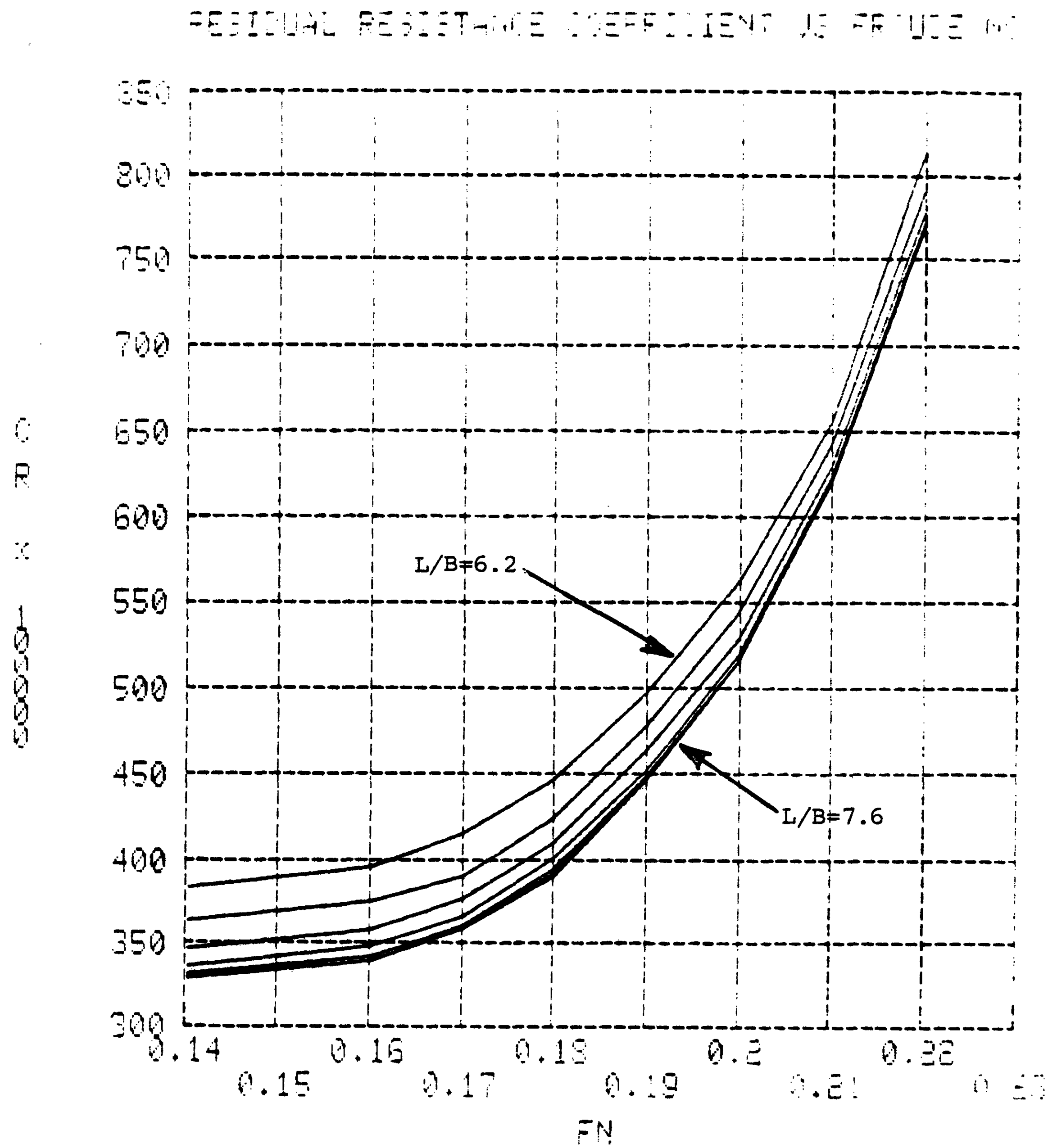


FIGURE 4.3

PLOT OF C_R VS. FN FOR C_B=0.805 AND B/T=2.76

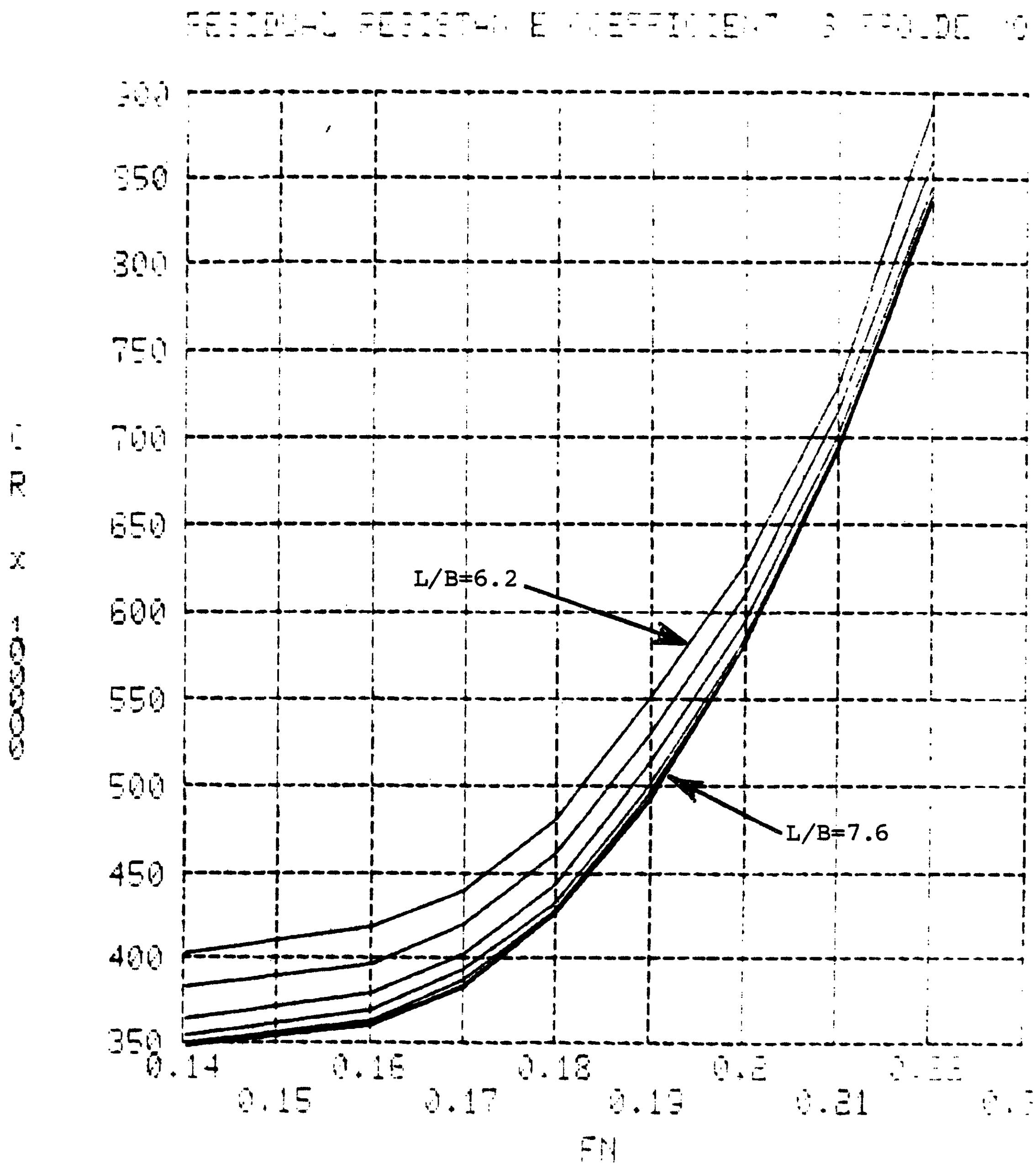


FIGURE 4.3

PLOT OF C_R VS. FN FOR C_B=0.815 AND B/T=2.76

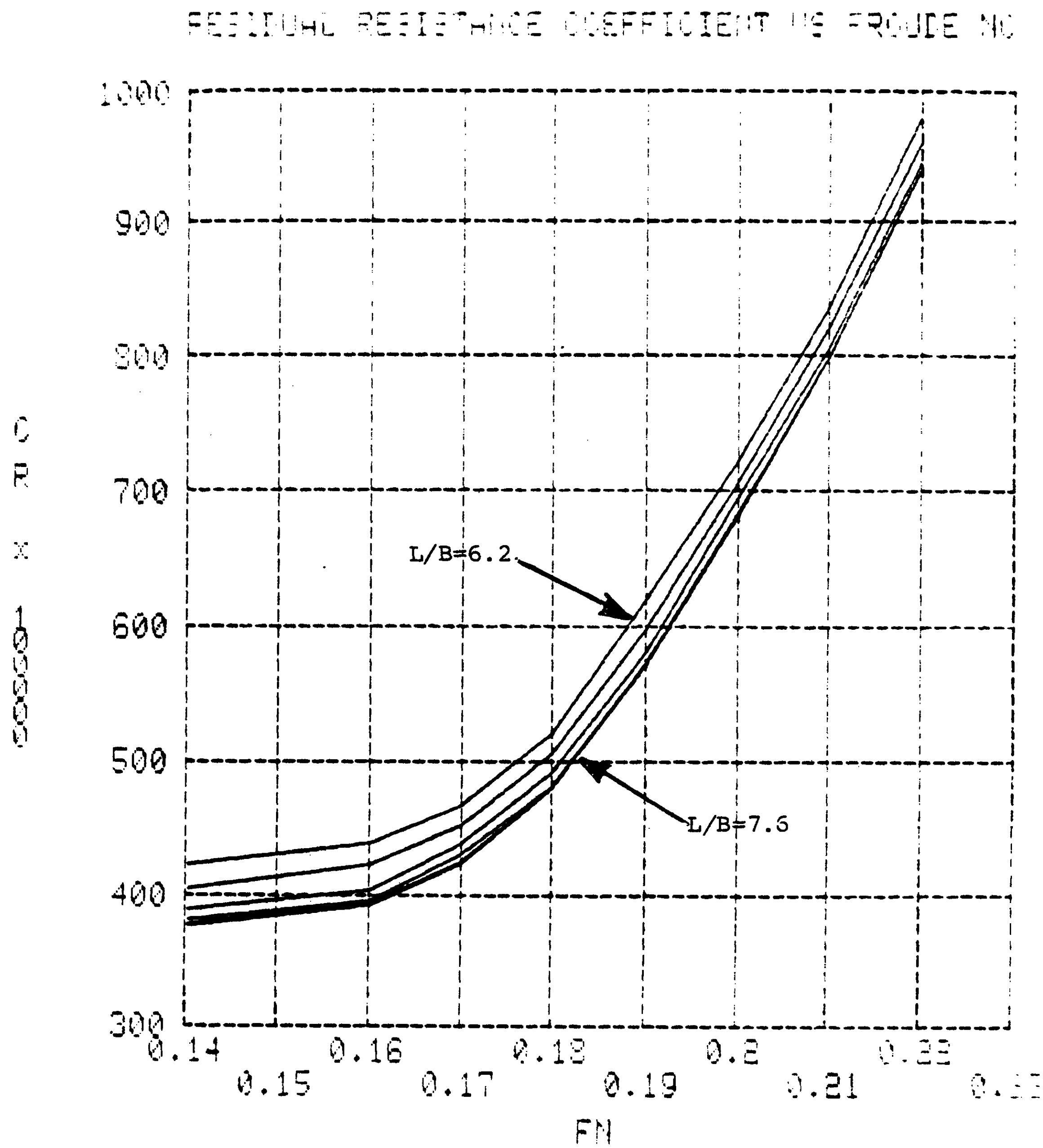


FIGURE 4.3

PLOT OF C_R VS. FN FOR C_B=0.825 AND B/T=2.76

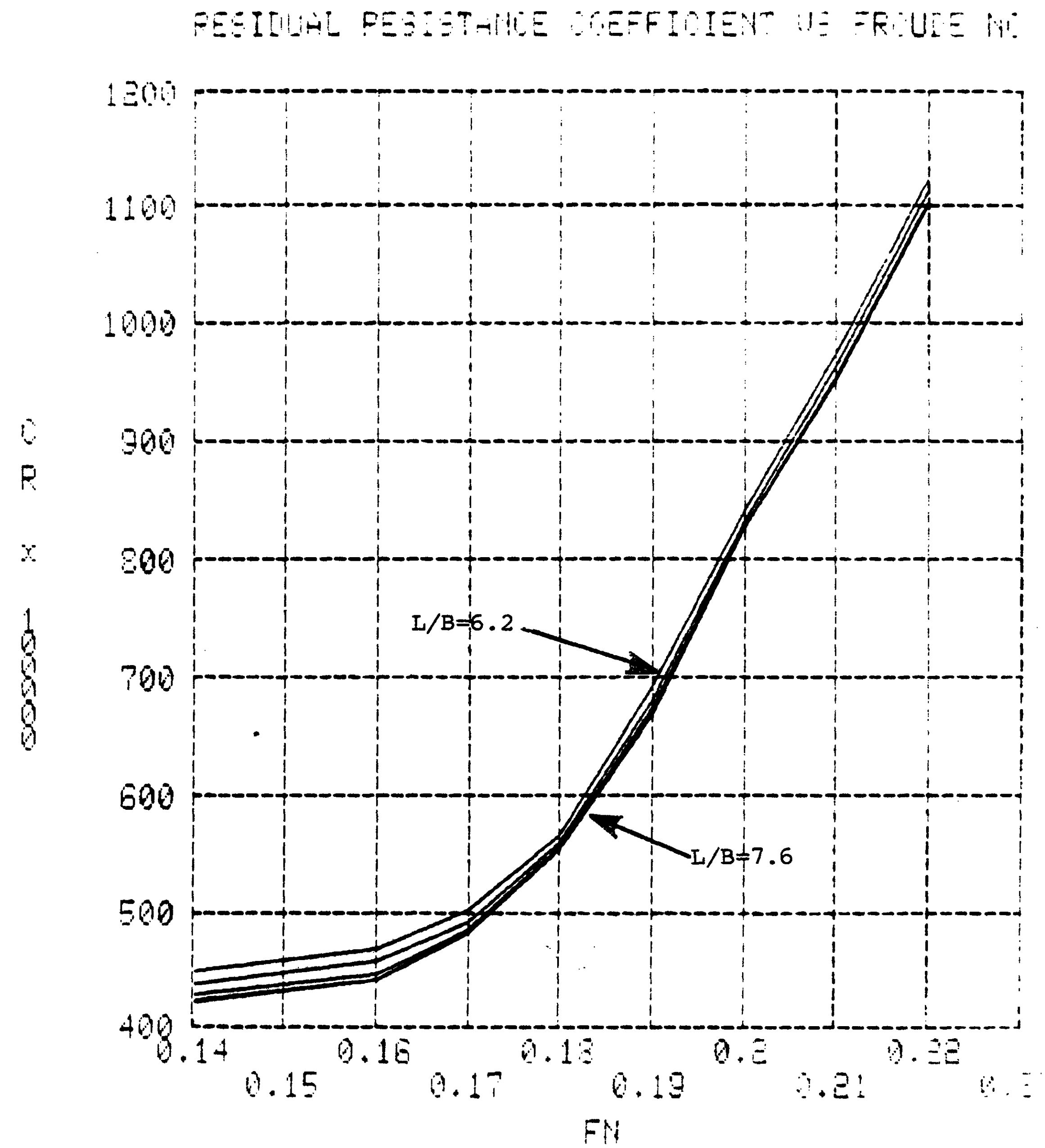


FIGURE 4.3

PLOT OF C_R VS. FN FOR C_B=0.835 AND B/T=2.76

TABLE 4.1

SELF PROPULSION FACTOR-W_T
WAKE FRACTION $((1-W_T)\times 1000)$

Block Coefficient	6.2	6.3	6.4	6.5	6.6	6.7	6.8	6.9	7.0	7.1	7.2	7.3	7.4	7.5	7.6
0.770	543	550	558	565	572	580	587	594	601	608	615	622	630	637	644
0.775	540	547	554	561	568	575	582	589	596	603	610	617	624	631	638
0.780	536	543	550	557	563	570	577	584	591	598	605	612	619	626	633
0.785	533	540	547	553	560	566	572	579	586	592	599	606	613	619	626
0.790	530	536	542	548	555	561	568	574	580	587	593	600	606	612	619
0.795	526	532	538	544	550	557	563	569	575	581	588	594	600	606	612
0.800	523	529	535	540	546	552	558	564	570	576	582	588	594	600	606
0.805	520	526	531	536	542	548	553	559	565	570	576	582	588	593	599
0.810	518	523	528	532	538	543	549	555	560	565	570	576	582	587	592
0.815	516	520	525	529	533	538	543	549	554	560	565	570	576	581	586
0.820	515	518	522	526	530	534	539	543	549	554	560	565	570	575	580
0.825	514	517	520	523	527	531	535	539	544	549	553	558	563	568	573
0.830	513	515	517	520	523	527	531	535	539	543	548	553	558	562	567
0.835	512	513	515	518	521	524	528	531	535	538	542	546	551	556	560
0.840	511	512	513	515	518	521	524	527	530	534	538	541	545	550	554

TABLE 4.1

SELF PROPULSION FACTOR-t

THRUST DEDUCTION FRACTION ((1-t)x1000)

Block Coefficient	6.2	6.3	6.4	6.5	6.6	6.7	6.8	6.9	7.0	7.1	7.2	7.3	7.4	7.5	7.6
0.770	769	772	775	778	782	784	787	790	793	797	800	803	806	809	812
0.775	769	772	775	778	781	784	787	790	793	796	799	802	805	808	811
0.780	769	772	775	778	781	784	787	790	793	796	799	802	805	808	811
0.785	769	772	775	777	781	784	787	790	792	796	798	801	804	807	810
0.790	768	771	774	777	780	783	786	789	792	795	798	801	804	807	810
0.795	768	771	774	777	780	783	786	788	792	795	797	800	803	806	809
0.800	768	770	773	776	779	782	785	788	791	794	797	800	802	805	808
0.805	767	770	773	776	778	782	785	787	790	793	796	799	802	805	807
0.810	766	769	772	775	778	781	784	787	789	792	795	798	801	804	807
0.815	765	768	771	774	777	780	782	785	788	791	794	797	800	802	806
0.820	764	767	770	773	775	778	781	784	787	790	793	795	798	801	804
0.825	763	765	768	771	774	777	780	782	785	788	791	794	797	800	802
0.830	761	763	766	769	772	775	778	780	783	786	789	792	795	797	800
0.835	758	761	764	767	770	773	775	778	781	784	787	789	792	795	798
0.840	755	759	761	764	767	770	772	775	778	781	784	786	789	792	795

TABLE 4.1

SELF PROPULSION FACTOR-η_R

RELATIVE ROTATING EFFICIENCY (X1000)

Block Coefficient	6.2	6.3	6.4	6.5	6.6	6.7	6.8	6.9	7.0	7.1	7.2	7.3	7.4	7.5	7.6
0.770	998	1003	1007	1010	1013	1016	1017	1017	1017	1016	1015	1013	1011	1008	1004
0.775	1000	1005	1009	1013	1016	1018	1019	1020	1020	1019	1018	1016	1014	1011	1007
0.780	1003	1007	1011	1015	1018	1021	1022	1023	1023	1022	1021	1019	1017	1014	1010
0.785	1005	1009	1014	1018	1021	1023	1024	1025	1025	1025	1024	1022	1020	1017	1013
0.790	1007	1012	1016	1020	1023	1026	1027	1028	1028	1028	1027	1025	1023	1020	1015
0.795	1009	1014	1018	1022	1026	1028	1030	1031	1031	1031	1030	1028	1026	1023	1018
0.800	1011	1016	1021	1025	1028	1031	1033	1033	1034	1033	1032	1031	1029	1025	1021
0.805	1013	1018	1023	1027	1031	1034	1035	1036	1037	1036	1035	1034	1032	1028	1024
0.810	1015	1020	1025	1030	1034	1036	1038	1039	1039	1039	1038	1037	1035	1031	1027
0.815	1017	1022	1027	1033	1036	1039	1041	1042	1042	1042	1041	1039	1037	1034	1030
0.820	1020	1025	1030	1035	1039	1042	1043	1044	1045	1045	1044	1042	1040	1037	1033
0.825	1022	1027	1032	1037	1041	1044	1046	1047	1048	1048	1047	1045	1043	1040	1036
0.830	1024	1029	1035	1040	1043	1046	1048	1050	1050	1050	1050	1048	1046	1043	1039
0.835	1026	1031	1037	1042	1046	1049	1051	1053	1053	1053	1052	1051	1049	1046	1042
0.840	1028	1033	1039	1045	1048	1051	1054	1055	1056	1056	1055	1054	1052	1048	1045

TABLE 4.2
FULL-BODIED SINGLE-SCREW RESIDUAL RESISTANCE COEFFICIENTS
FOR BREADTH/DRAFT = 2.46 (x 10000)

Froude Number	Block Coefficient	Length/Breadth							
		6.2	6.4	6.6	6.8	7.0	7.2	7.4	7.6
0.14	0.775	291	271	257	248	240	235	233	232
	0.785	323	303	286	273	264	258	254	252
	0.795	352	330	310	295	284	276	272	270
	0.805	368	349	328	312	298	288	284	282
	0.815	381	365	342	327	313	301	294	291
	0.825	412	388	364	348	334	324	315	310
	0.835	463	435	409	390	375	367	360	359
0.16	0.775	295	280	265	251	243	239	237	237
	0.785	333	315	298	285	272	268	265	263
	0.795	361	341	322	307	294	287	283	282
	0.805	378	357	338	322	308	300	294	292
	0.815	395	373	352	337	323	313	308	306
	0.825	425	400	378	361	348	337	330	327
	0.835	481	453	429	411	398	387	379	374
0.17	0.775	309	294	279	266	257	252	249	248
	0.785	346	330	313	298	288	282	278	277
	0.795	379	360	340	322	310	303	301	300
	0.805	398	378	357	338	326	318	316	315
	0.815	417	396	373	355	343	336	332	330
	0.825	452	428	405	387	373	366	360	358
	0.835	515	488	465	447	433	424	418	415
0.18	0.775	338	322	306	292	284	280	278	278
	0.785	371	353	338	322	313	309	307	306
	0.795	405	383	366	350	340	333	332	331
	0.805	433	409	389	372	360	353	351	350
	0.815	462	439	415	398	386	378	374	372
	0.825	509	483	460	440	427	419	414	412
	0.835	582	555	528	507	492	483	480	479

TABLE 4.2

FULL-BODIED SINGLE SCREW RESIDUAL RESISTANCE COEFFICIENTS

FOR BREADTH/DRAFT = 2.46 (x 1000)

Froude Number	Block Coefficient	Length/Breadth 6.2	6.4	6.6	6.8	7.0	7.2	7.4	7.6
0.19	0.775	380	363	347	334	328	323	322	322
	0.785	414	393	377	363	355	351	348	348
	0.795	449	426	406	392	382	377	374	373
	0.805	486	460	440	425	414	408	405	404
	0.815	531	507	484	469	457	450	447	446
	0.825	598	571	550	531	520	511	509	508
	0.835	696	665	641	620	608	599	591	590
0.20	0.775	448	428	412	400	394	390	388	388
	0.785	472	453	434	423	416	412	410	409
	0.795	504	484	463	450	443	440	438	437
	0.805	549	530	509	491	482	479	477	477
	0.815	609	590	570	554	543	538	535	534
	0.825	700	680	660	644	633	626	623	621
	0.835	830	808	788	770	758	746	740	740
0.21	0.775	545	527	510	496	488	486	485	484
	0.785	567	549	530	518	510	507	505	505
	0.795	599	580	560	546	538	534	532	532
	0.805	645	624	607	591	582	579	578	578
	0.815	709	690	673	659	650	644	642	641
	0.825	810	790	773	758	748	741	740	740
	0.835	952	933	915	900	890	884	882	880
0.22	0.775	684	664	647	635	628	623	620	619
	0.785	71[illegible]	696	677	663	656	650	648	647
	0.795	752	731	712	697	688	683	682	681
	0.805	797	774	756	740	731	727	724	723
	0.815	859	838	818	801	793	788	788	788
	0.825	955	930	909	895	885	881	880	880
	0.835	1100	1075	1052	1032	1026	1025	1025	1025

TABLE 4.2

FULL-BODIED SINGLE-SCREW RESIDUAL RESISTANCE COEFFICIENTS

FOR BREADTH/DRAFT = 2.76 (x 10000)

Froude Number	Block Coefficient	Length/Breadth							
		6.2	6.4	6.6	6.8	7.0	7.2	7.4	7.6
0.14	0.775	292	273	261	253	247	243	242	241
	0.785	324	307	293	284	278	274	273	272
	0.795	355	338	323	313	307	304	303	303
	0.805	383	363	346	335	330	328	327	327
	0.815	402	382	363	353	348	347	347	347
	0.825	422	404	388	379	376	375	375	375
	0.835	448	437	427	422	420	420	420	420
0.16	0.775	302	284	271	263	257	253	251	250
	0.785	337	318	303	294	289	287	286	286
	0.795	368	347	332	323	317	313	312	312
	0.805	395	374	357	347	341	338	338	338
	0.815	418	396	378	368	362	360	359	359
	0.825	439	422	403	395	392	391	391	391
	0.835	468	457	445	440	440	440	440	440
0.17	0.775	313	297	284	277	273	272	271	270
	0.785	347	330	317	310	307	305	304	304
	0.795	383	362	349	340	335	333	333	333
	0.805	415	390	376	366	360	358	358	358
	0.815	440	420	402	392	386	383	382	382
	0.825	467	452	438	430	425	424	423	423
	0.835	502	492	485	483	482	482	482	482
0.18	0.775	340	325	312	303	300	298	297	297
	0.785	374	358	345	336	331	329	328	328
	0.795	410	391	378	368	362	360	359	359
	0.805	446	424	410	400	394	392	390	390
	0.815	481	462	443	433	428	427	426	426
	0.825	520	505	491	481	480	480	480	480
	0.835	566	560	556	554	554	554	554	554

TABLE 4.2
FULL-BODIED SINGLE-SCREW RESIDUAL RESISTANCE COEFFICIENTS
BREADTH/DRAFT = 2.76 (x 10000)

Froude Number	Block Coefficient	Length/Breadth							
		6.2	6.4	6.6	6.8	7.0	7.2	7.4	7.6
0.19	0.775	381	364	351	347	346	346	346	346
	0.785	414	399	385	378	377	377	377	377
	0.795	453	437	421	413	410	410	410	410
	0.805	497	478	463	452	448	447	447	447
	0.815	551	530	513	501	496	493	492	492
	0.825	618	596	580	572	570	570	570	570
	0.835	692	681	674	670	669	669	669	669
0.20	0.775	440	430	419	412	411	410	410	410
	0.785	472	460	448	440	438	437	437	437
	0.795	511	496	483	474	472	472	471	471
	0.805	561	543	528	519	515	515	515	515
	0.815	627	609	593	583	580	580	580	580
	0.825	721	706	694	683	680	680	680	680
	0.835	840	830	825	825	825	825	825	825
0.21	0.775	541	527	513	505	503	502	502	502
	0.785	568	553	540	532	530	530	530	530
	0.795	602	590	578	570	572	572	572	572
	0.805	655	642	630	623	620	620	620	620
	0.815	729	715	701	695	693	692	692	692
	0.825	835	820	805	799	798	798	798	798
	0.835	972	962	952	950	950	950	950	950
0.22	0.775	683	665	651	643	640	639	639	639
	0.785	717	699	686	679	677	677	677	677
	0.795	760	741	727	720	720	720	720	720
	0.805	813	791	779	772	769	769	769	769
	0.815	891	860	845	839	836	836	836	836
	0.825	980	960	945	940	940	940	940	940
	0.835	1123	1112	1103	1100	1100	1100	1100	1100

CHAPTER 5

DROP-AND-SWAP MODEL RESULTS, OBSERVATIONS, AND SUGGESTED REFINEMENTS

5.1 Base Case Results

At this point, the structure and details of the drop-and-swap model, including the barge powering and design subprograms, have been discussed. It is now time to consider what can be learned from the model. To do this, the model was run for five one-way trades with annual cargo flows of 100,000; 600,000; 1,000,000; 6,000,000; and 10,000,000 long tons. The user specified inputs and the values of the semi-fixed parameters used in these runs are shown in Table 5.1 and Figure 2.5, respectively. Essentially, in these base case runs the model finds the optimum barge size, speed, and form within the system parameter ranges specified in Table 2.1 for port pair trades with port separation distances of 500, 2000, 3500 and 5000 nautical miles (NM) and for a reasonable range of loading/discharging rates. In these base case trades, it is assumed there are no draft or beam restrictions. Barge length, however, is limited to 750 feet since the ABS

rules used in the barge design model are applicable only to barges less than this length. Additionally, barge size is limited to 100,000 DWT since this is the maximum size that designers have even considered for barge construction. (1) Finally, it is assumed that there are no costs associated with terminal loading/discharging or storage facilities. The value of the cargo is assumed to be $200 per long ton and the other cost parameters are as discussed in Chapter 2.

Graphical output from these runs of the required freight rate as a function of loading/discharging (L-D) rate for four port separation distances (500, 2000, 3500 and 5000 nautical miles) is shown in Figures 5.1-5.5. The printed outputs from these runs are voluminous and so cannot be included here. However, some of the results, including the range of optimum barge deadweights and tug speeds, have been extracted and presented in Table 5.2. In addition, some observations that can be made concerning all the runs are presented in the next subsection. This is followed by a discussion of points of interest that pertain to an individual run. And, this is followed by some conclusions that can be drawn from the whole series of base runs.

---

(1) The largest barge currently in operation is the Breit/Ingram tug-barge system Presque Isle of 53,000DWT operating on the Great Lakes. The largest ocean going barges will be the 47,000 DWT CATUG tank barges being built for Aramada Hess.

5.1.1 General Observations Pertaining to All Base Case Runs

The following general observations can be made after examining the printed output of the base case runs:

1. The optimum barge form for all the port pair trades has the smallest allowable length-breadth (6.0) and breadth-draft (2.0) ratios. These ratios result in the shortest barge with the least amount of hull steel for a given deadweight capacity. (1) Apparently, at the slow speeds that these tug-barge systems operate, the capital cost savings achieved by constructing short, blunt barges outweigh any cost penalties associated with the higher residual resistance of these forms.

2. The optimum block coefficient for all the port pair trades varies from 0.78 to 0.81. The 0.78 value is usually associated with tug-barge units advancing at eleven knots or faster while the 0.81 value is usually found with systems advancing at nine knots or slower. However, for small tug-barge units, less than 15,000 DWT, a block coefficient of as low as 0.78 may be found for speeds as low as nine knots.

3. Port pair trades with longer port separation distances and faster L-D rates usually have optimum systems with larger barges.

4. Port pair trades with longer port separation distances and lower L-D rates usually have optimum systems with a

(1) This result can be verified by examining the elasticity exponents of the regression equations presented in Table 3.13.

greater number of tug and barge units.

Observations that are peculiar to an individual base case and not already summarized in Table 5.2 are presented in the following four subsections.

5.1.2 Base Case Results: Annual Cargo Flows of 100,000 LT

When annual cargo flows are as low as in this case, all of the cargo can usually be moved less expensively in a single small barge operating in the ship-like integral mode. The drop-and-swap mode is not competitive due to the extra barge or barges needed to be stationed in port. As can be seen from Figure 5.1, the only time that the drop-and-swap mode can be of advantage is when L-D rates are very low (less than 2000 LT/day) in trades with long port separation distances. In these trades, the extensive in port time forces integral mode operations to require more than one tug unit to handle the annual cargo flows. This makes integral mode operation uncompetitive compared to drop-and-swap mode operation with only one tug.

5.1.3 Base Case Results: Annual Cargo Flows of 600,000 and 1,000,000 LT

Both of these cases are similar in that for port pair trades with low L-D rates, the drop-and-swap mode will be less expensive. Conversely, for those trades having high L-D rates, the integral mode will be less expensive. As seen in

Figures 5.2 and 5.3, the major difference between these cases is that for the 1,000,000 LT case, the L-D rate tradeoff points are usually larger than for the 600,000 LT case. It was observed from both cases that for a given port separation distance, the L-D rate tradeoff point between the two modes usually occurs when the optimal integral mode begins to require only one more tug than the optimal drop-and-swap mode. Then, above the tradeoff L-D rate, the extra cost of the barges needed to remain in port for drop-and-swap mode operation begin to outweigh the capital cost savings resulting from the one fewer tug.

5.1.4 Base Case Results: Annual Cargo Flows of 6,000,000 LT

As seen in Figure 5.4, for this case the drop-and-swap mode will be the most economical method of operation for all port pair trades with port separation distance greater than 3500 NM. This is because with large cargo flows, the savings resulting from economies of scale push the optimum barge size to the upper constraint of 100,000 DWT. Due to this constraint, the optimum number of integral tug-barge units cannot be reduced (at economical speeds of operation) sufficiently fewer than the number of drop-and-swap tugs to make the integral mode economically competitive. Thus, when barge size is constrained, the higher tug utilization at a given speed achieved by the drop-and-swap mode of operation outweighs any barge capital cost savings inherent in the inte-

gral mode. For the shorter port separation distance (500 and 2000 nautical miles) the barge size is not binding so that the integral mode does become more economical, but at comparatively high L-D rates. It should be noted that when barge size is not binding the optimal system speeds vary between seven and eleven knots, However, when the barge size approaches the upper limit, the optimum integral mode speeds become as high as thirteen knots.

5.1.5 Base Case Results: Annual Cargo Flows of 10,000,000 LT

In this case the barge size constraint causes the drop-and-swap mode to be the operating mode of choice for all port pair trades except for some with port separation distance of 500 NM. For those trades, the integral mode will be favored when L-D rates exceed 45,000 LT/day. Again, it should be noted that barge size constraints have forced the optimum integral mode systems to operate with speeds of up to fifteen knots.

5.1.6 Conclusions From Base Case Runs

After reviewing all the ouput from the base case runs, a few general conclusions can be drawn concerning tug-barge systems operating on a port pair trade in either the drop-and-swap or integral mode. The major conclusion is that port pair trades can be broken up into three groups, primarily based on the amount of annual cargo flow and secondarily based

on the port separation distance. These groups are 1) trades for which the integral mode will be the operating method of choice for all L-D rates, 2) trades for which the drop-and-swap mode will be the operating method of choice for all L-D rates, and 3) trades for which the drop-and-swap mode will be optimum for low L-D rates and the integral mode will be optimum for high L-D rates.

The first group consists of trades with annual cargo flow requirements less than the ton-mile capacity of a single small (less than 25,000 DWT) tug-barge unit operating at an economical speed (6 to 9 knots). In these trades, since all the cargo can be easily transported in a single tug-barge unit, the higher tug utilization available from drop-and-swap operation is unnecessary. Thus, the integral mode will always be the operating method of choice in these trades.

The second group consists of trades with annual cargo flow requirements much greater than the ton mile capacity of two or more tug-barge units of maximum carrying capacity. The constraint on barge size prevents full use of economies of scale in the integral mode of operation. This, in turn, prevents operation in the integral mode at economical speeds with sufficiently less capital equipment to make up for the increased tug utilization savings inherent in the drop-and-swap mode of operation. Thus, the drop-and-swap mode will always be the operational mode of choice in deadweight constrained trades.

The third group consists of trades not falling within the first two groups. That is, trades for which annual cargo flows are too large to be carried in one tug-barge unit but too small to be carried in several maximum sized units. It is for these trades that there will be a L-D tradeoff point below which the drop-and-swap mode and above which the integral mode will be the operational method of choice.

In general, for trades with greater annual cargo flows and longer port separation distances, this tradeoff point is at greater L-D rates. This is because these trades demand more ton-mile transport capacity which can be met by either increasing the number or size of the tug-barge units. If the number of units is increased, then the additional barges that must remain in port in the drop-and-swap operations become less significant. If the barge size is increased, the increase for the drop-and-swap mode operation will be less than for integral mode operations due to the higher tug utilization efficiency. In either case the drop-and-swap mode is favored in these trades more than the integral mode.

It should be mentioned that trades with very short port separation distances (i.e. 500 NM) have L-D tradeoff points at higher values than for longer distance trades. This is because the percentage of the total voyage time spent in port (about 40% at the L-D tradeoff point) is large for these trades, favoring the higher tug utilization efficiencies achieved by drop-and-swap mode of operation.

From the base case runs, certain conclusions can also be drawn concerning the form and speed of the optimum barge. It appears that since the Froude number is so low (less than 0.16) at the optimum tug-barge speeds of seven to eleven knots that residual resistance does not have much significance. Thus, optimum barge forms have the greatest draft and breadth possible to reduce the length; and, consequently, the barge capital cost. This saving outweighs any resistance penalty caused by the blunt barge forms. The only concession made for the effect of resistance is with respect to block coefficient. For tug-barge speeds greater than ten knots finer lines are required while blunter lines are suitable at slower speeds.

5.2 Sensitivity Runs

Sensitivity runs were made to see the effect on required freight rates and L-D rate tradeoff points of changes in some of the semi-fixed parameter values used in the base case runs. Specifically, changes to the value of the cargo, the shoreside storage costs, and the maximum barge draft were investigated.

Due to the cost of these runs, an exhaustive set, including a wide range of variation of a single parameter or combination of parameters, could not be made. However, the runs whose inputs are given in Table 5.3 should give an indication, although not conclusive, of the effects of changes to their values. In the three sections that follow, an analysis is made of the output from the sensitivity runs.

5.2.1 Sensitivity Run: Change in Cargo Value

A sensitivity run was made to see what effects a change in the cargo value from $200 to $0 per long ton would have on the 1,000,000 LT annual cargo flow base case. As seen in Figure 5.6 and Table 5.4, this reduction in cargo value reduces the required freight rate by less than $.20 for trades with port separation distance of 500 NM to over $1.00 for trades with a port separation distance of 5000 NM. As to be expected, the reduction is in proportion to the port separation distance, which is indicative of the sea time and the time value of the cargo. It was also noticed that in the base case run, that the optimum system speeds would sometimes be one, two, or even three knots faster than the sensitivity run. This confirms what is to be expected, that higher cargo values result in higher optimum system speeds. Thus, it is seen that unless the time value of cargo is included in any transportation system optimization model, the transport vehicles will be optimized at too slow a speed.

5.2.2 Sensitivity Runs: Changes to Shoreside Storage Costs

Two sensitivity runs were made to determine what effects the inclusion of shoreside storage capital construction costs of $48 per long ton storage capacity (1) would have on the

(1) This value for storage capacity capital costs was obtained from a major oil company and is indicative of oil tankage construction costs for large tanks as of January 1979. Also, the model assumes that the amount of shoreside storage capacity

1,000,000 and 6,000,000 long ton annual cargo flow base cases. In addition, a run was made to determine the effect on the 1,000,000 long ton base case when a $1.00 per long ton storage facility throughput charge was also included.

It would be expected that since shoreside storage costs are applicable to the integral mode only, the inclusion of such costs would favor the drop-and-swap mode of operation. This expected result is confirmed by the graphical and tabular output of these runs presented in Figures 5.7-5.9 and Table 5.4. From both cases it is seen that the effect of the capital cost of shoreside storage on the integral mode becomes more pronounced with longer port separation distances. This is because these port pair trades demand the larger barges and consequently, larger and more expensive storage tanks. (1) Also, the effect of storage is greater in trades with low L-D rates since these trades require more shoreside loading and discharging terminal facilities to handle the annual cargo flows.

The combined results of both of these effects is that the L-D tradeoff point is shifted to higher values, especially for trades with longer port separation distances and that the

---

for a given port pair trade is equal to the product of the number of terminal facilities in the port times the deadweight of the optimum barge operating in the integral mode for that port pair trade.

(1) For the 1,000,000 LT annual cargo flow trades, the present valued annual capital charges for storage amount to approximately $0.0564 per 10,000 DWT tankage and LT of cargo flow.

advantage the drop-and-swap mode has for trades with low L-D rates becomes more pronounced.

The addition of the $1.00 storage operating cost favors the drop-and-swap mode even more than the storage capital costs. This is because it essentially increases all the integral mode trades' rfr by $1.00, equivalent to the annualized capital cost for over 175,000 DWT of tankage. As seen by comparing Figures 5.8 and 5.9, this results in a shift of the integral mode curves by $1.00 so that the L-D tradeoff point moves to a much higher value. For the port pair trades with a separation distance of 500 NM, where the integral mode has rfr's only marginally lower than the drop-and-swap mode at high L-D rates, the inclusion of the operating storage costs increases the integral mode rfr so much that the drop-and-swap mode dominates for all L-D rates. Since the differences between the drop-and-swap and integral rfr curves is more pronounced with large port separation distance, the tradeoff point shift is not as great for these trades. However, it is apparent that if shoreside storage costs are included in the economic analysis, the trades for which the drop-and-swap mode will be optimal will be extended to those of significantly higher loading/discharging rates.

5.2.3 Sensitivity Runs: Inclusion of Port Draft Limit

A sensitivity run was made to investigate the effects that a draft limitation would have on the 1,000,000 LT annual

cargo flow base case. Additionally, another run was made to see if the same effect occurred to the 1,000,000 LT trades that also included a $48 per long ton storage terminal capital cost. The graphical and tabular results of these runs are shown in Figures 5.10 and 5.11 and Table 5.4.

Comparing these figures with the unrestricted draft cases shown in Figures 5.3 and 5.7, it was observed that there was little effect on trades which were optimized with small barges, those of less than 450 feet or 25,0000 DWT. However, the effect was very pronounced in those trades which in the unrestricted draft case would be optimized with large barges. In those trades, both the drop-and-swap and integral mode rfr's were increased by several dollars. This is because the draft restrictions forced the optimum feasible barges of a given deadweight capacity to have greater lengths and consequently, greater capital cost. Additionally, since barge length was constrained to be less than 750 feet, the maximum barge size was limited to less than 65,000DWT. This forced those trades which were optimized in the unrestricted draft cases to obtain the necessary ton-mile capacity by either increasing the speed of the tugs or the number of tug-barge units in operation--both requiring the rfr to increase even further. The draft limitation affects both the drop-and-swap and integral mode trades. However, since the majority of unlimited draft port pair trades are optimized with larger barges in the integral than the drop-and-swap mode, the effect

is somewhat greater for the integral mode. Thus, the L-D trade-off point is shifted to somewhat higher values in the longer distance trades where larger barges are predominant. Nevertheless, the shift is not very great.

Generally, it is seen that port draft restrictions are very costly in trades which have large annual cargo flows and/or long port separation distances and thus would normally take advantage of the economies of scale inherent in large barges. These trades, however, are slightly more favorable to the drop-and-swap mode of operation.

5.3 Suggestions for Model Refinement and Future Research

The simplest refinement that can be made to the drop-and-swap model is to verify and calibrate the barge hull steel weight regression equations obtained from the barge design subprogram with hull weight data from single-skin tank-barges currently in operation. This requires only the acquisition of detailed design data (especially block coefficient, percentage high strength steel used, scantling length, and hull steel weight) for a series of barges of different size and form. However, since this data is of proprietary nature, it was not obtained.

Another major refinement could be achieved by using tank-barge resistance and propulsion data instead of the single-screw, full-bodied data that was used in the powering subprogram. The tank-ship data, although adequate for the

preliminary economic anlaysis accomplished in the drop-and-swap model, is probably not accurate enough to be used for preliminary design purposes. However, other than results of some towing tank model tests presented in Robinson (1976) there is no published literature on the hydrodynamic performance of large push-towed tug-barge forms, single or twin screw. Therefore, there is a definite need for a series of towing tank model tests of tug-barge forms similar to those currently in use. Suggested ranges of forms that should investigated are length-breadth ratios from 4.0 to 7.0 and breadth-draft ratios from 2.0 to 4.0 and block coefficients from 0.80 to 0.90. Evaluation is particulary needed of the drag caused by current and proposed linkage designs and by draft differentials between the tug and barge.

Another refinement could be achieved by updating, calibrating, and validating the tug capital cost regression equations. This would require researching tug machinery and linkage costs, and possibly developing a preliminary tug design subprogram. However, the latter may not be necessary if sufficient tug and supply boat data can be obtained to develop reliable regression equations for tug hull and outfit weights and/or costs.

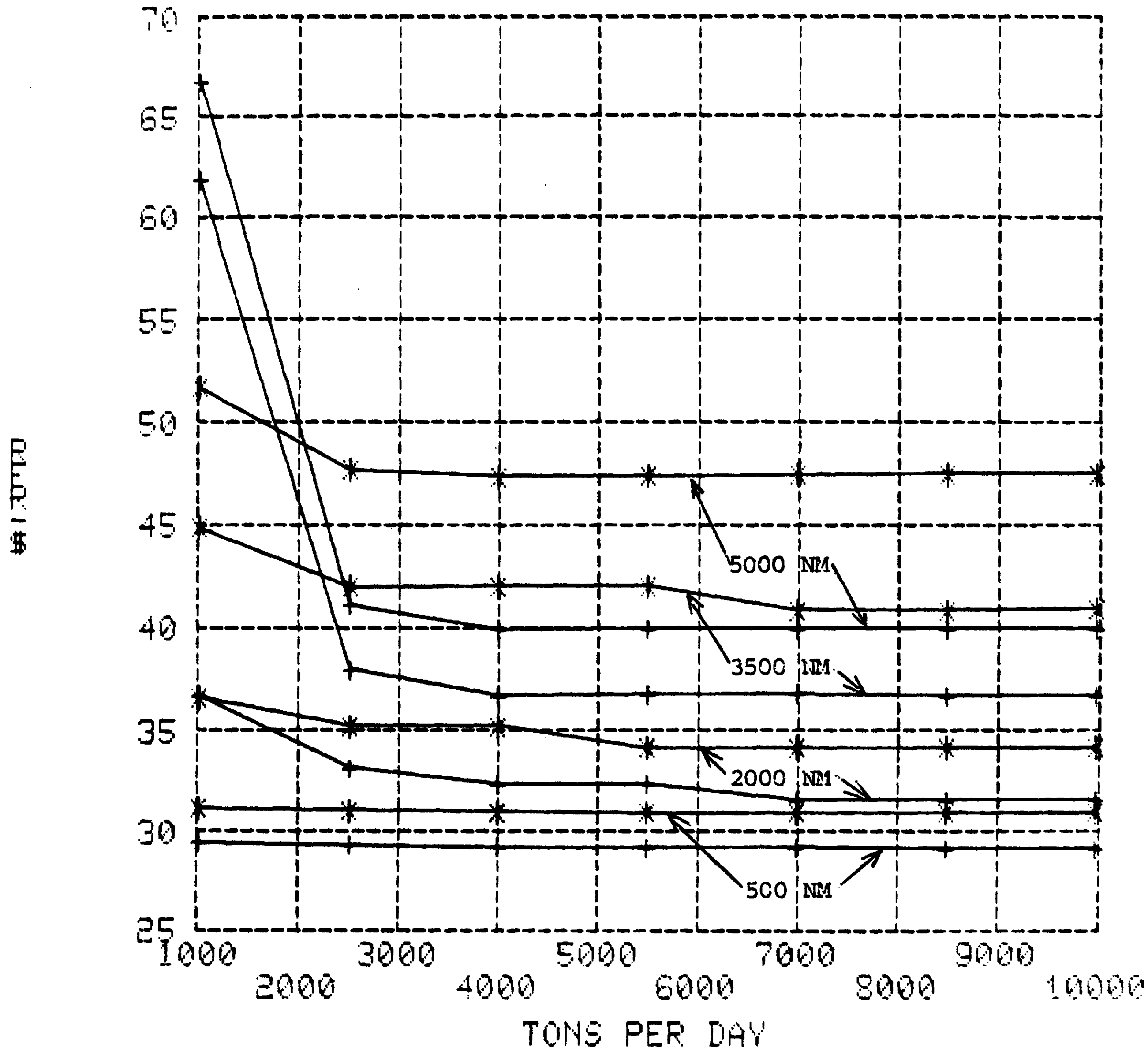
The model could also be improved by taking into account the stochastic nature of shipping operations. Tug-barge systems can not operate on precise time schedules on account of delays due to weather, equipment malfunction, strikes, port

congestion, etc. Since the drop-and-swap model is deterministic, and only uses average process times, it can not simulate the unpredictable nature of tug-barge operations. Nevertheless, for systems with a large number of tugs and barges, the stochastic nature will not be very important due to system averaging. Only in the small systems might these effects be of concern. Therefore, it would be useful to develop a simulation model that incorporates stochastic operations for a small system, say one tug and three barges. A comparison of the stochastic model results with that from the deterministic drop-and-swap model would then indicate how important is the stocahstic nature of the system.

The stochastic nature of shipping schedules would also have an important effect on the amount of shoreside storage that would be needed for the integral mode of operation. However, given an estimate of the stochastic parameters of the system which could be obtained from a simulation model, the amount of shoreside storage can be determined from basic inventory theory.

Finally, the drop-and-swap model could be improved by extending it to multiport trades. Such an extension would complicate the problem significantly, especially if barges of different sizes can be used. Most likely network theory would need to be utilized to find the optimum tug-barge systems. However, since the stochastic nature of the problem would probably be very significant in multiport cases where

scheduling is of utmost importance, it may be that only simulation models could be used for multiport trades.



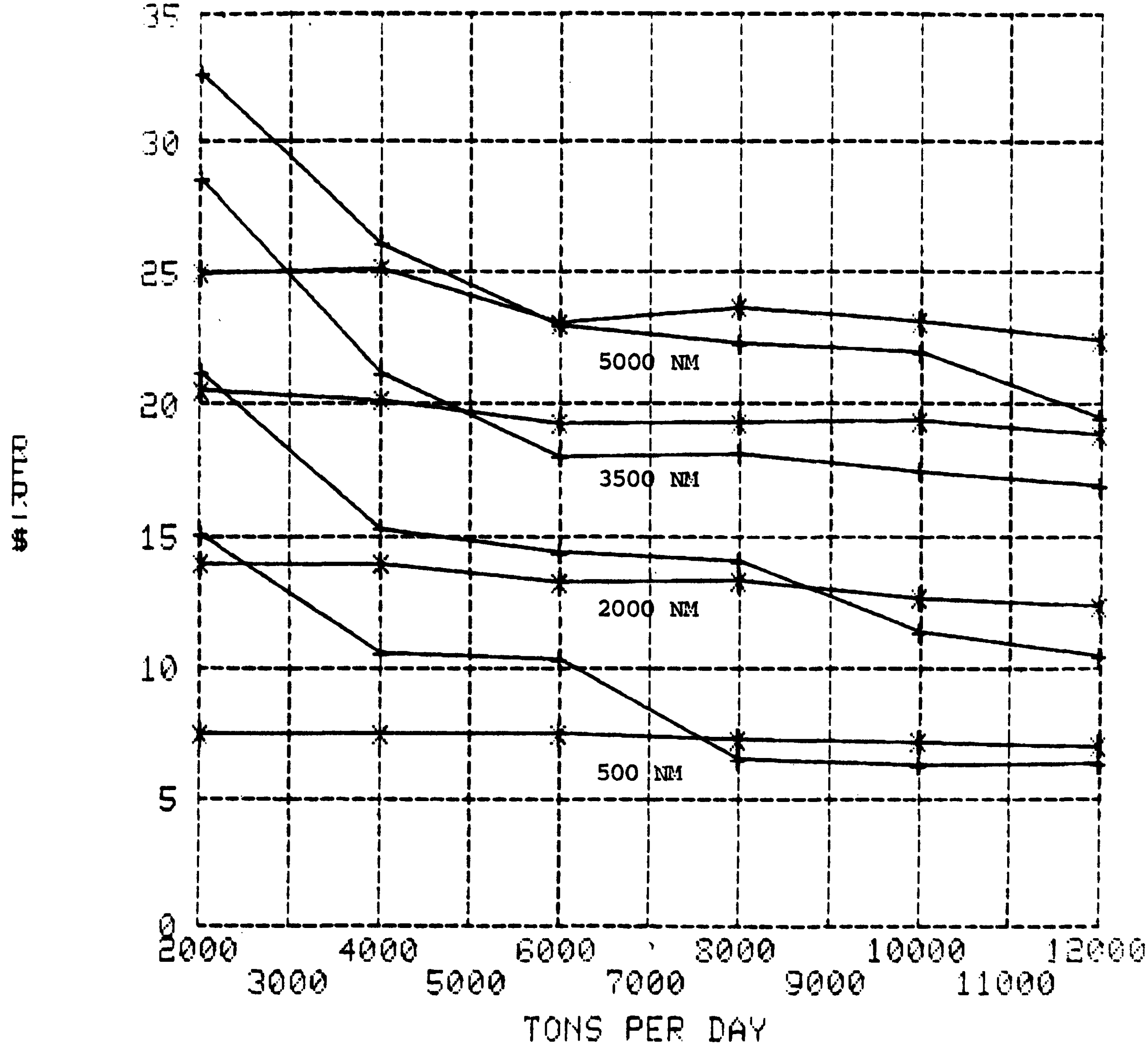
KEY: ✱ -- DROP-AND-SWAP MODE

+ -- INTEGRAL MODE

FIGURE 5.1

PLOT OF RFR VS. TERMINAL L-D RATE

BASE CASE RUN: ANNUAL CARGO FLOW OF 100,000 LT



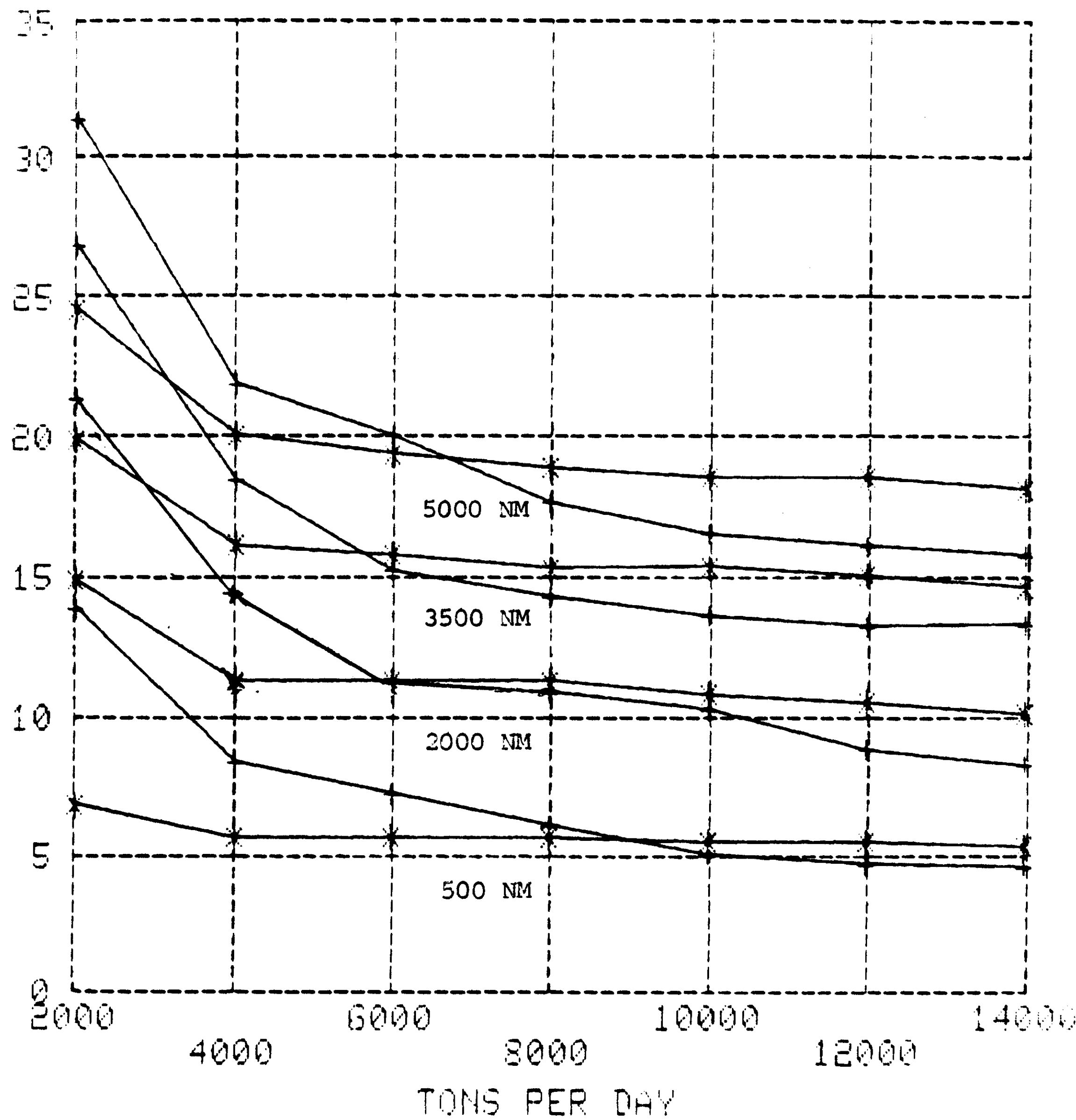
KEY: ✱ -- DROP-AND-SWAP MODE

+ -- INTEGRAL MODE

FIGURE 5.2

PLOT OF RFR VS. TERMINAL L-D RATE

BASE CASE RUN: ANNUAL CARGO FLOW OF 600,000 LT



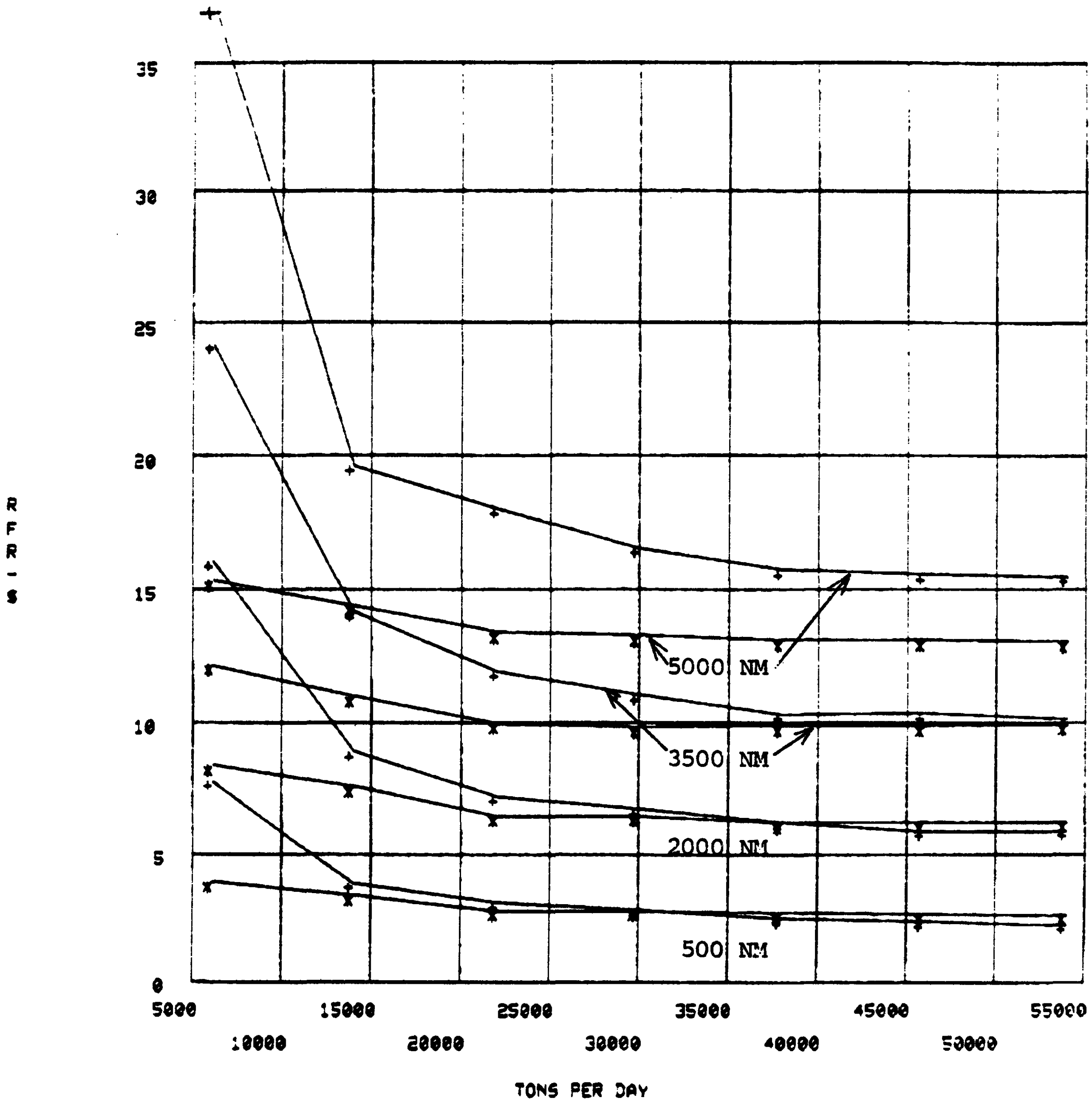
KEY: ✗ -- DROP-AND-SWAP MODE

+ -- INTEGRAL MODE

FIGURE 5.3

PLOT OF RFR VS. TERMINAL L-D RATE

BASE CASE RUN: ANNUAL CARGO FLOW OF 1,000,000 LT



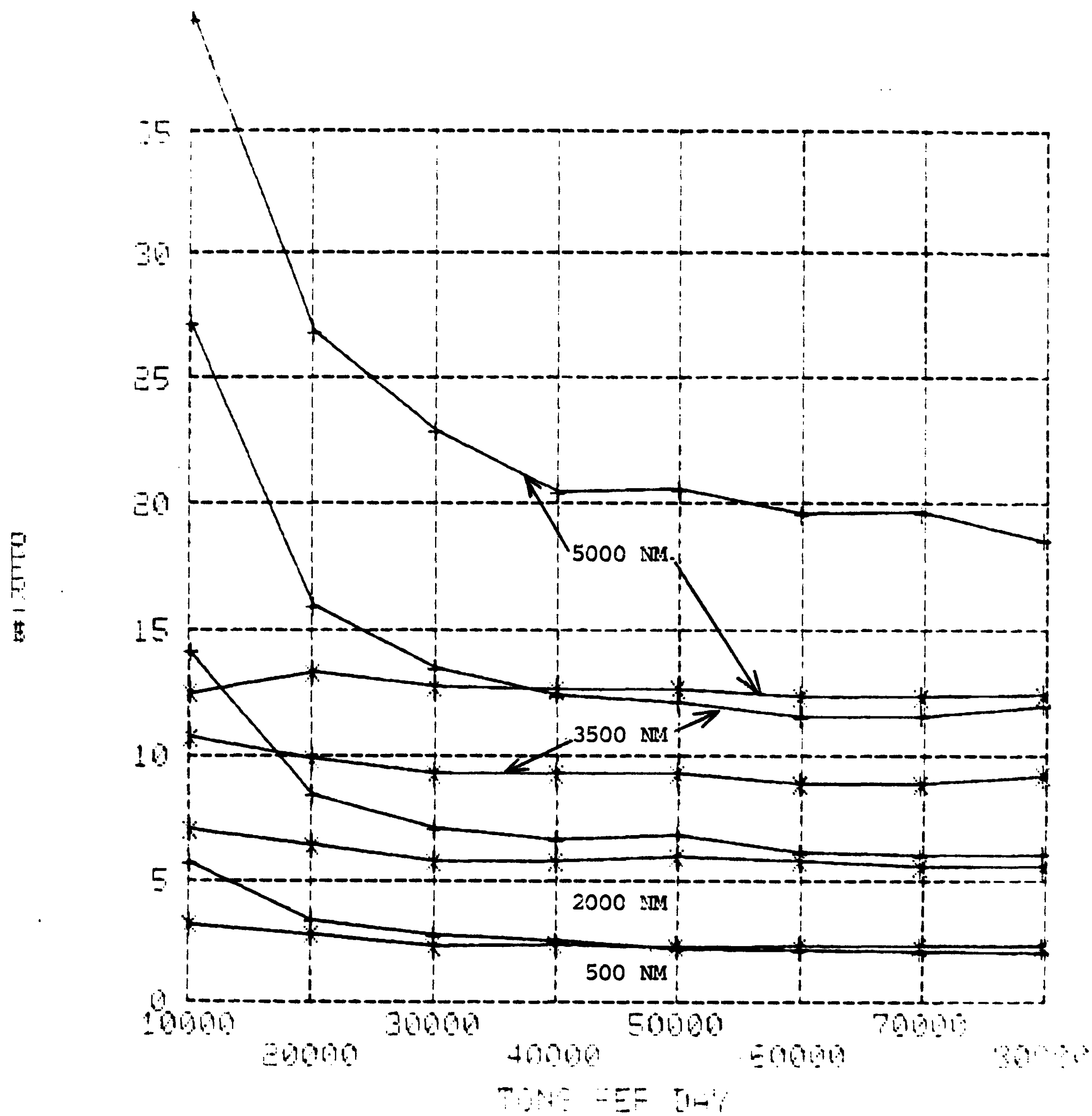
KEY: X -- DROP-AND-SWAP MODE

+ -- INTEGRAL MODE

FIGURE 5.4

PLOT OF RFR VS. TERMINAL L-D RATE

BASE CASE RUN: ANNUAL CARGO FLOW OF 6,000,000 LT



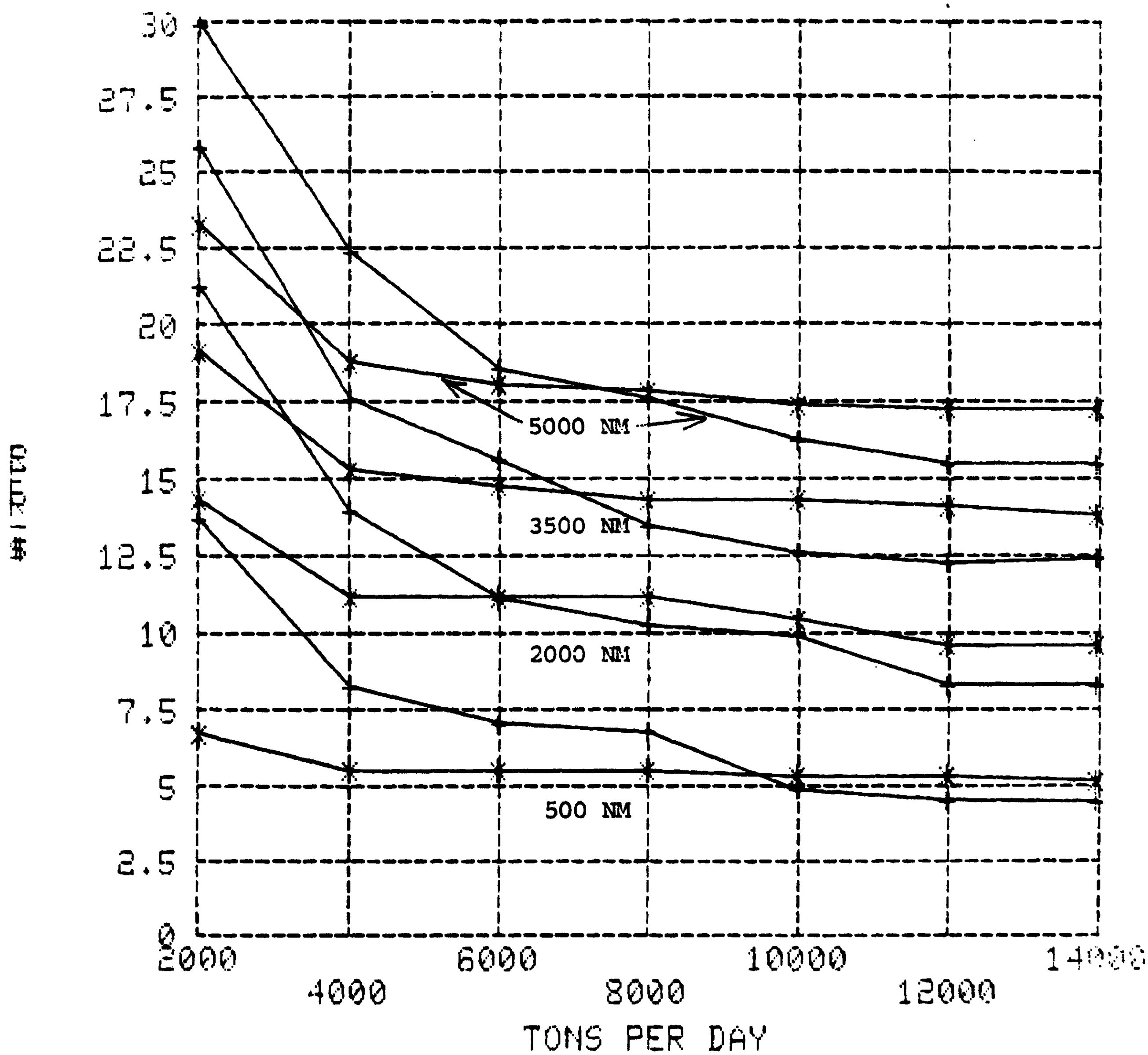
KEY: ✗ -- DROP-AND-SWAP MODE

+ -- INTEGRAL MODE

FIGURE 5.5

PLOT OF RFR VS. TERMINAL L-D RATE

BASE CASE RUN: ANNUAL CARGO FLOW OF 10,000,000 LT



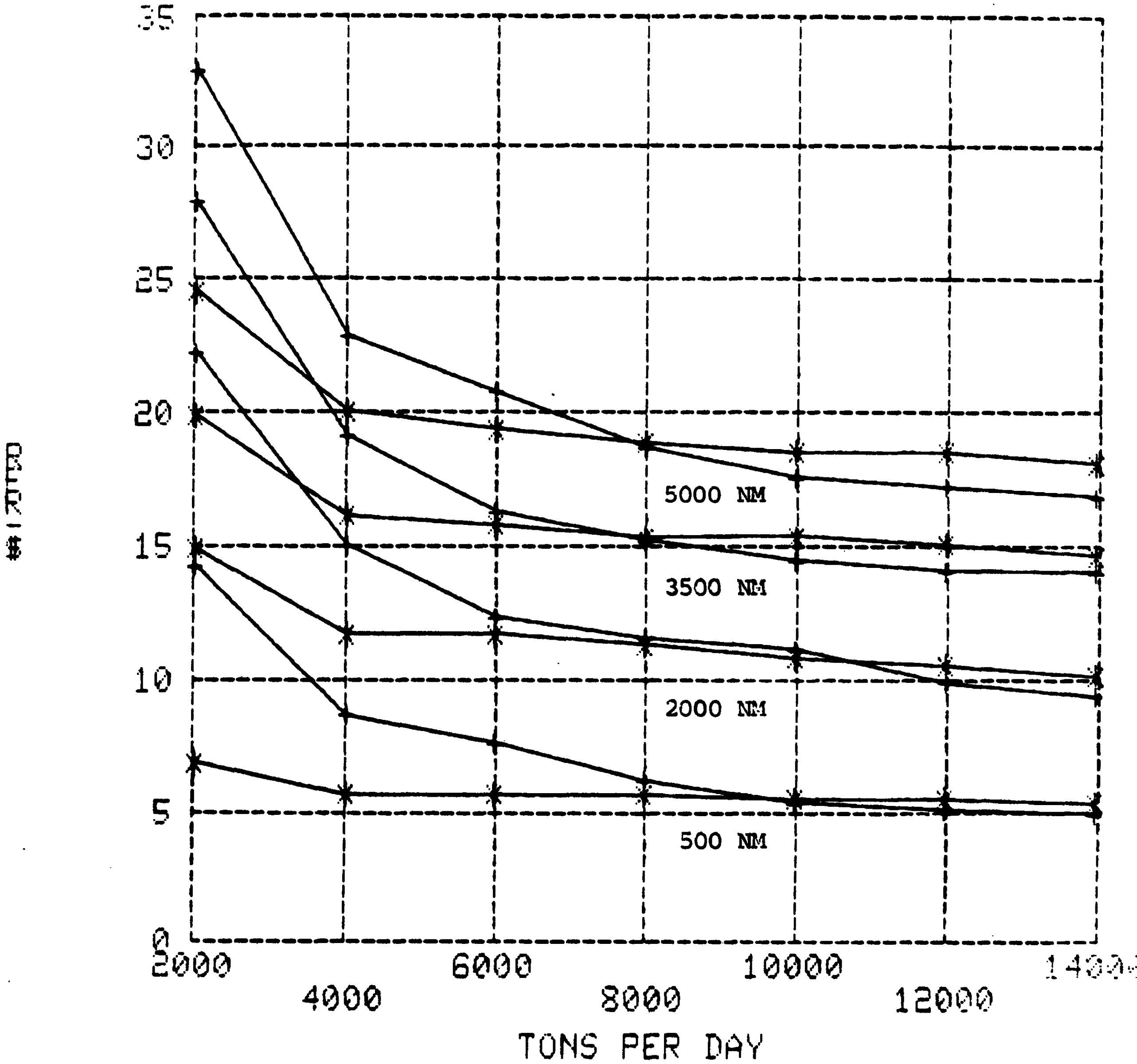
KEY: ᛯ -- DROP-AND-SWAP MODE

\+ -- INTEGRAL MODE

FIGURE 5.6

PLOT OF RFR VS. TERMINAL L-D RATE

SENSITIVITY RUN: ANNUAL CARGO FLOW OF 1,000,000 LT
WITH CARGO VALUE OF $0

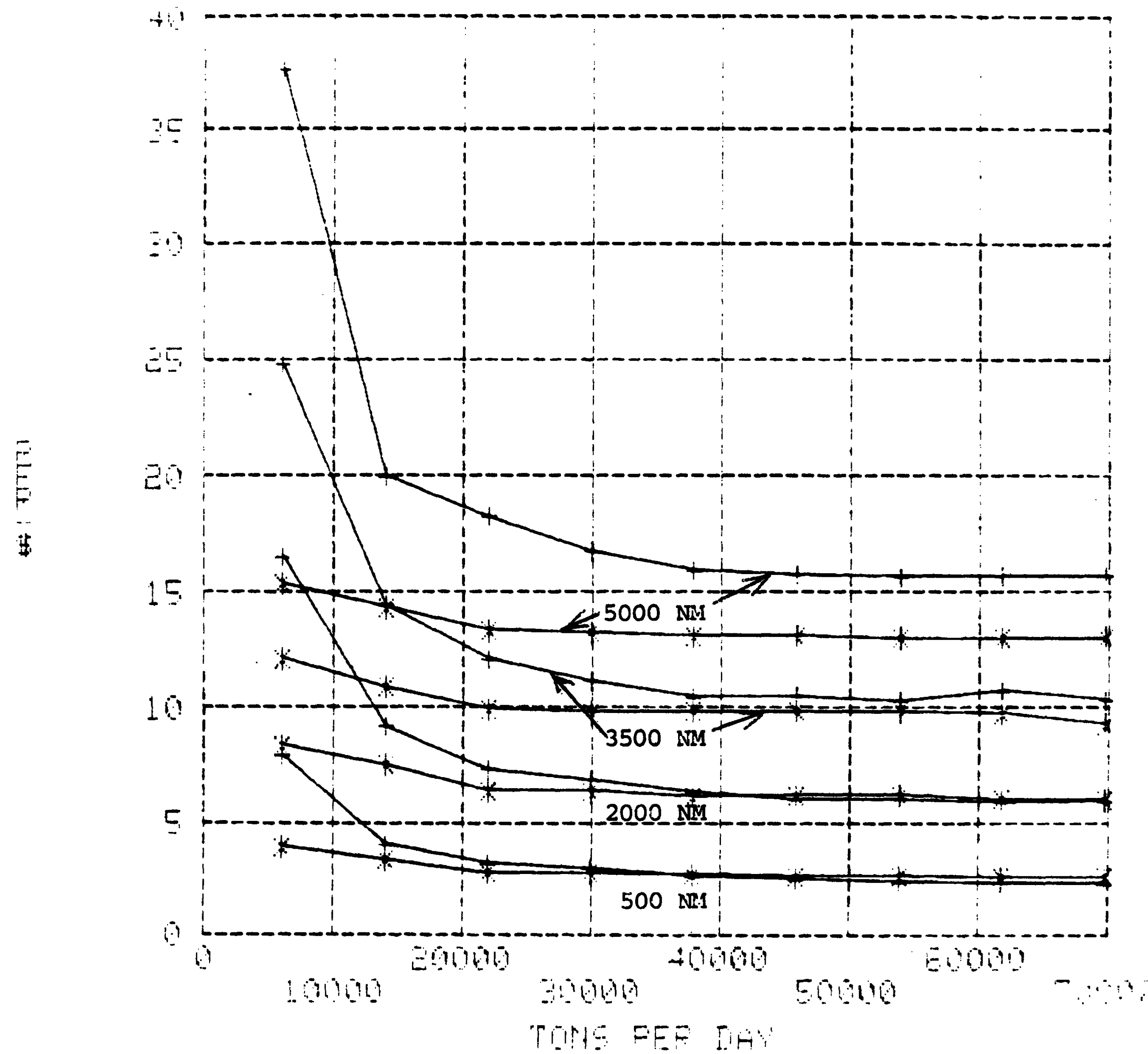


KEY: ✗ -- DROP-AND-SWAP MODE
+ -- INTEGRAL MODE

FIGURE 5.7

PLOT OF RFR VS. TERMINAL L-D RATE

SENSITIVITY RUN: ANNUAL CARGO FLOW OF 1,000,000 LT
WITH STORAGE FACILITY CAPITAL COST OF $48/LT



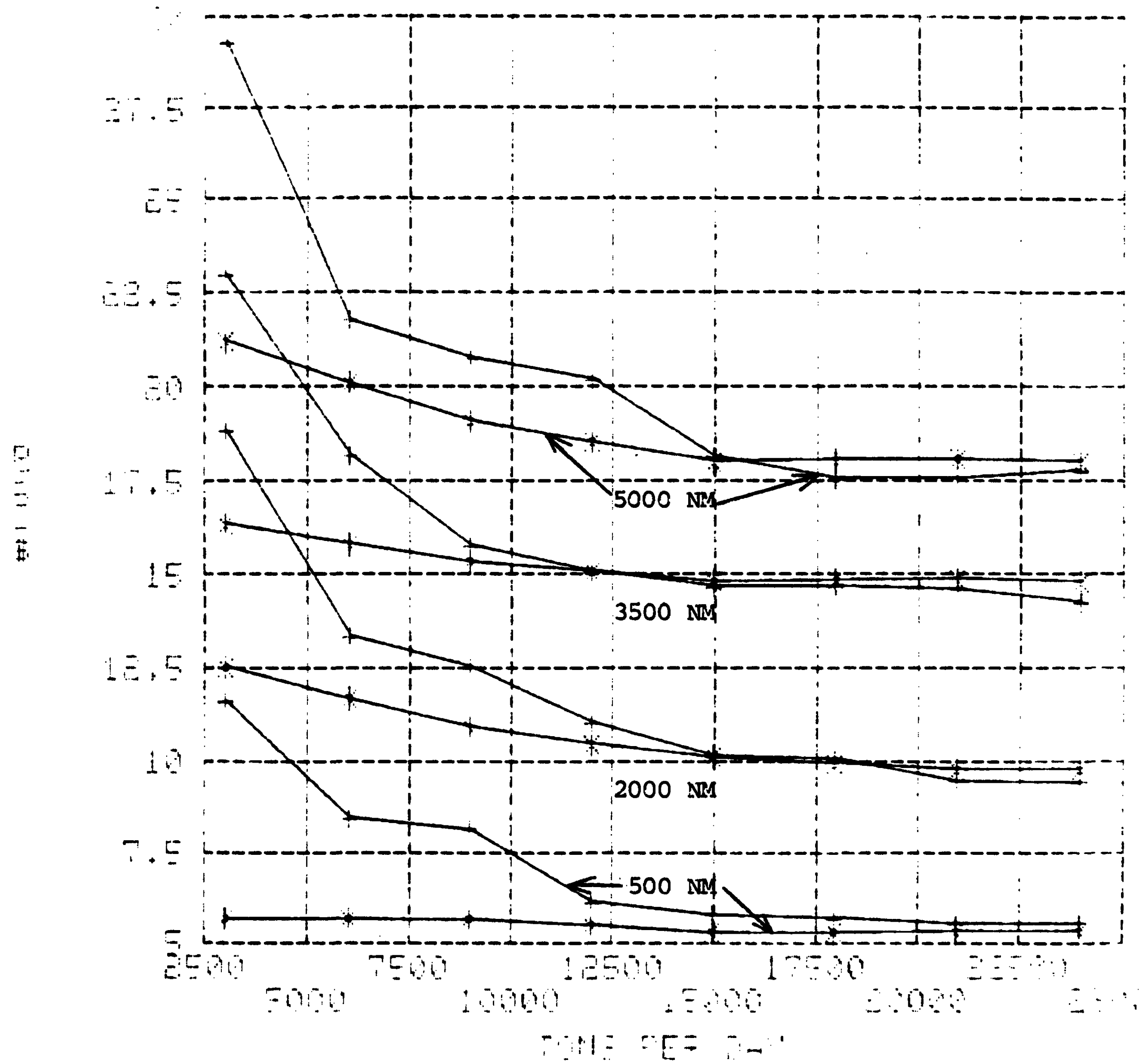
KEY: ⋇ -- DROP-AND-SWAP MODE

+ -- INTEGRAL MODE

FIGURE 5.8

PLOT OF RFR VS. TERMINAL L-D RATE

SENSITIVITY RUN: ANNUAL CARGO FLOW OF 6,000,000 LT
WITH STORAGE FACILITY CAPITAL COST OF $48/LT



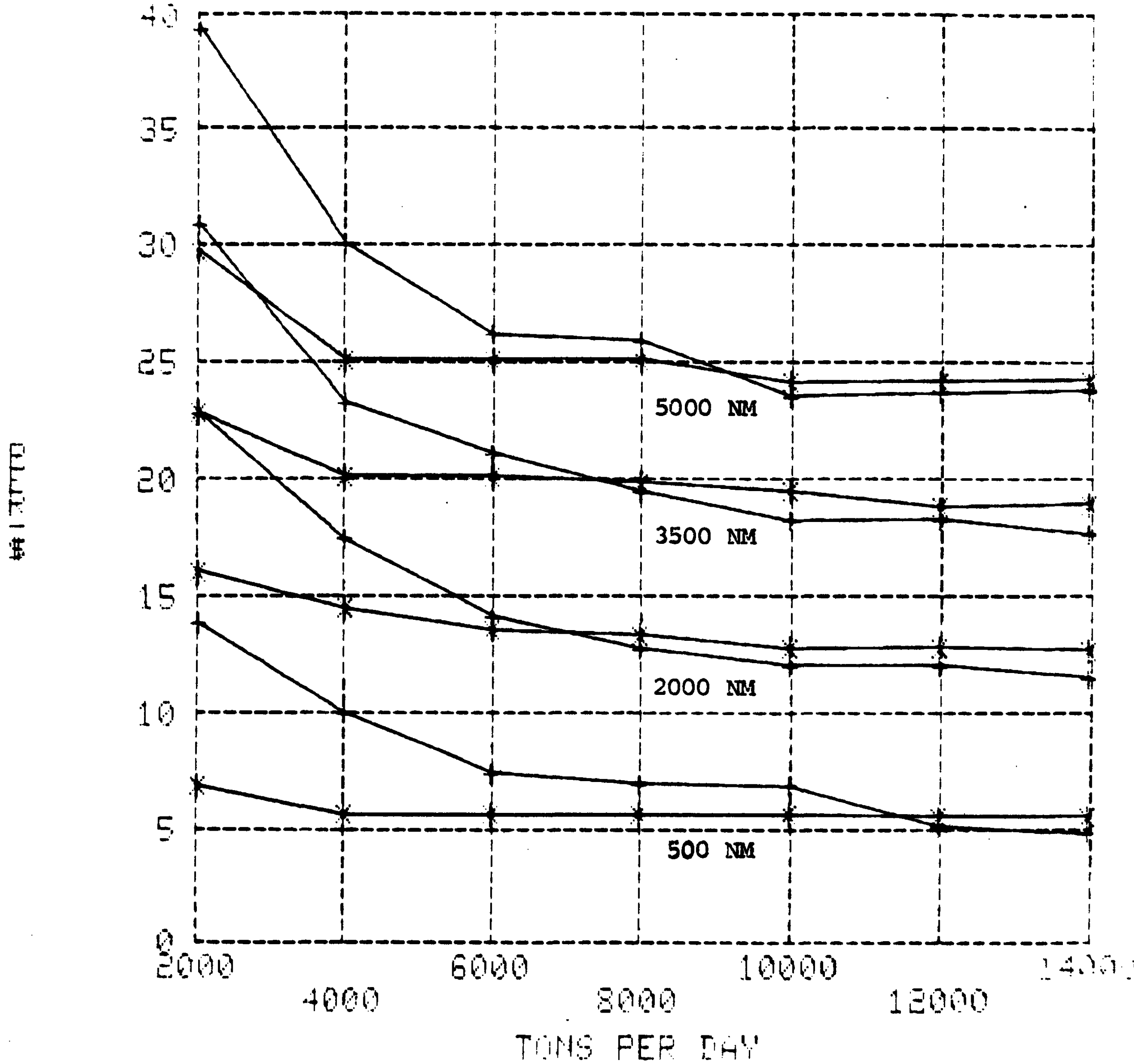
KEY: ✗ -- DROP-AND-SWAP MODE

\+ -- INTEGRAL MODE

FIGURE 5.9

PLOT OF RFR VS. TERMINAL L-D RATE

SENSITIVITY RUN: ANNUAL CARGO FLOW OF 1,000,000 LT
WITH STORAGE FACILITY CAPITAL COST OF $48/LT
WITH STORAGE FACILITY OPERATING COST OF $1/LT

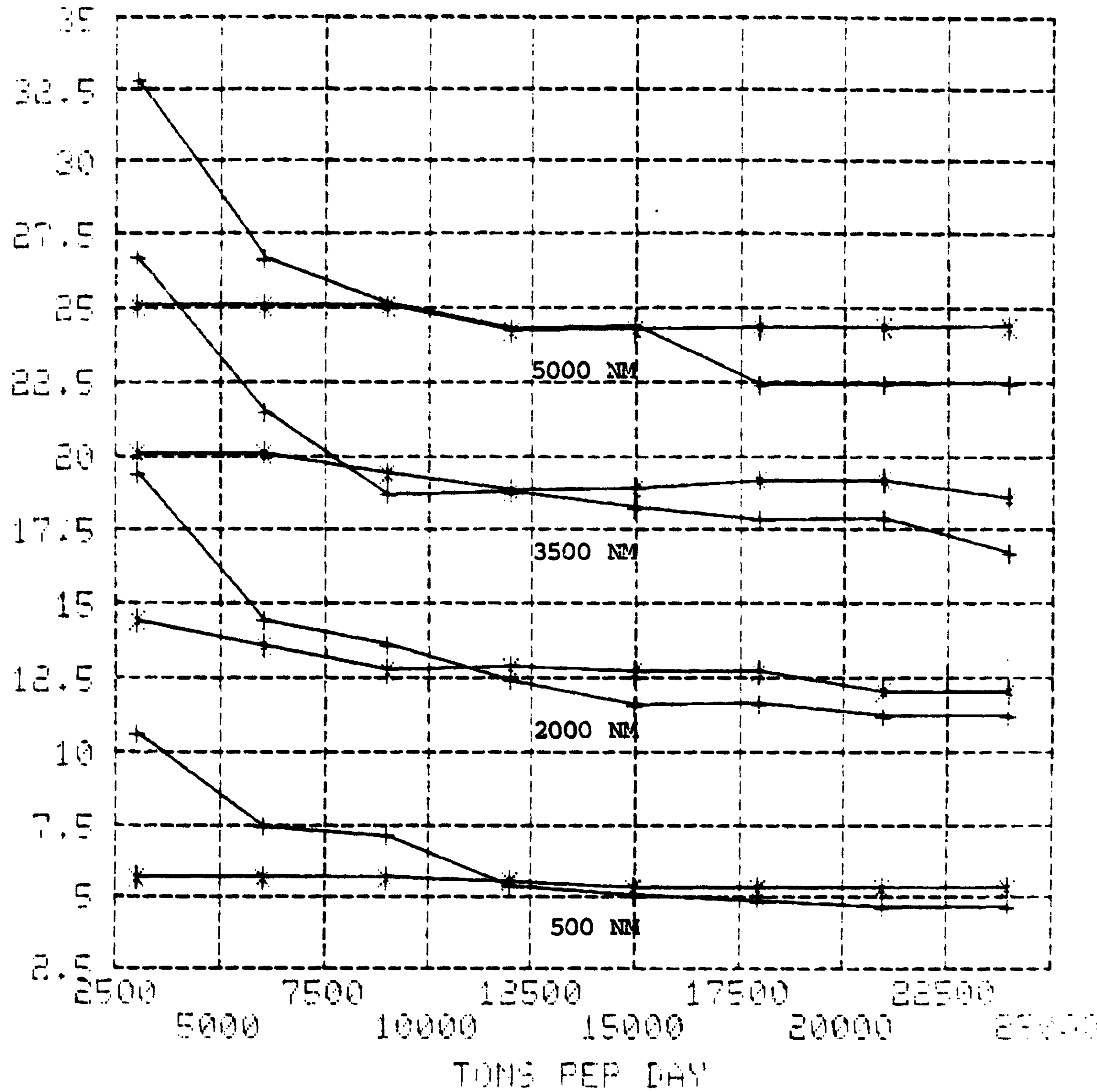


KEY: ✗ -- DROP-AND-SWAP MODE
+ -- INTEGRAL MODE

FIGURE 5.10

PLOT OF RFR VS. TERMINAL L-D RATE

SENSITIVITY RUN: ANNUAL CARGO FLOW OF 1,000,000 LT
WITH BARGE DRAFT LIMITED TO 38'



KEY: ✗ -- DROP-AND-SWAP MODE

+ -- INTEGRAL MODE

FIGURE 5.11

PLOT OF RFR VS. TERMINAL L-D RATE

SENSITIVITY RUN: ANNUAL CARGO FLOW OF 1,000,000 LT
WITH BARGE DRAFT LIMITED TO 38'
WITH STORAGE FACILITY CAPITAL COST OF $48/LT

TABLE 5.1

SUMMARY OF INPUTS USED IN BASE CASE RUNS

aflowave1 aflowave2	minrate maxrate delrate	mindist maxdist deldist	mindwt maxdwt deldwt	mincb maxcb delcb	minlb maxlb dellb	minbt maxbt delbt	minspeed maxspeed delspeed	Changes to Semifixed Parameters
100,000	1,000	500	5,000	0.77	6.0	2.0	6.0	None
0	10,000	5,000	50,000	0.84	6.6	3.0	12.0	
	1,500	1,500	2,500	0.01	0.2	0.5	1.0	
600,000	2,000	500	5,000	0.77	6.0	2.0	6.0	None
0	12,000	5,000	70,000	0.84	6.6	3.0	13.0	
	2,000	1,500	2,500	0.01	0.2	0.5	1.0	
1,000,000	2,000	500	5,000	0.77	6.0	2.0	6.0	
0	14,000	5,000	100,000	0.84	6.6	3.0	13.0	None
	2,000	1,500	5,000	0.01	0.2	0.5	1.0	
6,000,000	6,000	500	5,000	0.77	6.0	2.0	7.0	
0	54,000	5,000	100,000	0.84	6.6	3.0	14.0	None
	8,000	1,500	5,000	0.01	0.2	0.5	1.0	
10,000,000	10,000	500	30,000	0.77	6.0	2.0	8.0	
0	80,000	5,000	100,000	0.84	6.6	3.0	16.0	None
	10,000	1,500	5,000	0.01	0.2	0.5	1.0	

TABLE 5.2

TABULAR OUTPUT SUMMARY FOR BASE CASE RUNS

Port Separation Distance (Nautical miles)	OUTPUT PARAMETER	a f l o w a v e l (LT)				
		100,000	600,000	1,000,000	6,000,000	10,000,000
500	D&S Mode Speed Range (Kts)	6	7-9	7-10	8-10	8-10
	Integral Mode Speed Range (Kts)	6	6-9	6-12	7-10	8-10
	D&S Mode DWT Range(1000 LT)	5	10-15	15-25	30-70	55-90
	Integral Mode DWT Range (1000 LT)	5	7.5-20	15-40	40-70	60-85
	D&S Integral Mode L-D Rate Tradeoff Point (LT/day)	none: Integral mode always favorable	Between 6000 & 8000	Between 8000 & 10000	Between 30,000 & 38000	Between 40,000 & 50,000
	%Porttime is of Total Voyage Time at L-D Rate Tradeoff Point	--	44-31	72-59	45-39	37-38
2000	D&S Mode Speed Range (Kts)	7-8	8-12	8-10	8-9	9-10
	Integral Mode Speed Range (Kts)	6-9	7-12	7-12	9-11	10-12
	D&S Mode DWT Range (1000 LT)	7.5	32.5-42.5	30-80	75-95	85-100
	Integral Mode DWT Range (1000 LT)	7.5 - 12.5	27.5-42.5	50-95	75-100	85-100
	D&S Integral Mode L-D Rate Tradeoff Point(LT/day)	near 1000	Between 8000 & 10,000	Near 6000	Near 38000	None: D&S Mode always favorable
	%Porttime is of Total Voyage Time at L-D Rate Tradeoff Point	57	35-22	49	23	--
3500	D&S Mode Speed Range (Kts)	10-13	8-9	8-11	9-10	9
	Integral Mode Speed Range (Kts)	6-9	8-12	8-9	10-12	12-13
	D&S Mode DWT Range (1000 LT)	10 - 12.5	32.5-37.5	40-100	95-100	90-100
	Integral Mode DWT Range (1000 LT)	10 - 12.5	27.5-37.5	50-100	95-100	90-100
	D&S Integral Mode L-D Rate Tradeoff Point (LT/day)	Between 1000 & 2500	Between 4000 & 6000	Between 4000 & 6000	None: D&S Mode always favorable	None: D&S Mode always favorable
	%Porttime is of Total Voyage Time at L-D Rate Tradeoff Point	29	43-29	48	--	--
5000	D&S Mode Speed Range (Kts)	8-10	8-10	8-10	9-10	9-12
	Integral Mode Speed Range (Kts)	6-7	8-11	8-10	10-13	13-15
	D&S Mode DWT Range (1000 LT)	12.5-17.5	37.5-50	45-90	90-100	60-100
	Integral Mode DWT Range (1000)	12.5-22.5	37.5-52.5	70-1000	90-100	60-100
	D&S Integral Mode L-D Rate Tradeoff Point (LT/day)	Between 1000 & 2500	Near 6000	Between 6000 & 8000	None: D&S Mode always favorable	None: D&S Mode always favorable
	%Porttime is of Total Voyage Time at L-D Rate Tradeoff Point	29	29	36-32	--	--

TABLE 5.3

SUMMARY OF INPUTS USED IN SENSITIVITY CASE-RUNS

aflowave1 aflowave2	minrate maxrate delrate	mindist maxdist deldist	mindwt maxdwt deldwt	mincb maxcb delcb	minlb maxlb dellb	minbt maxbt delbt	minspeed maxspeed delspeed	Changes to Semixed Parameter
1,000,000	2,000	500	5,000	0.78	6.0	2.0	6.0	vcargo = 0
0	14,000	5,000	100,000	0.82	6.0	2.0	13.0	
	2,000	1,500	5,000	0.01			1.0	
6,000,000	6,000	500	30,000	0.78	6.0	2.0	7.0	cfixstor = 48
0	70,000	5,000	100,000	0.81	6.0	2.0	15.0	
	8,000	1,500	5,000	0.01			1.0	
1,000,000	4,000	500	5,000	0.78	6.0	2.0	6.0	cfixstor = 48
	16,000	5,000	100,000	0.81	6.0	2.0	13.0	
	2,000	1,500	5,000	0.01			1.0	
1,000,000	3,000	500	5,000	0.78	6.0	2.0	6.0	cfixstor = 48
0	24,000	5,000	100,000	0.81	6.0	2.0	13.0	cvarstor = 1
	3,000	1,500	5,000	0.01			1.0	
1,000,000	3,000	500	5,000	0.78	6.0	2.0	5.0	cfixstor = 48
0	24,000	5,000	100,000	0.81	6.6	3.0	12.0	maxt1 = maxt2 = 38
	3,000	1,500	5,000	0.01	0.2	0.5	1.0	
1,000,000	2,000	500	5,000	0.75	6.0	2.0	6.0	maxt1 = maxt2 = 38
0	14,000	5,000	100,000	0.83	6.4	3.0	13.0	
	2,000	1,500	5,000	0.01	0.2	0.5	1.0	

TABLE 5.4

TABULAR SUMMARY FOR SENSITIVITY RUNS

Port Separation Distance (Nautical Miles)	OUTPUT PARAMETERS	SENSITIVITY RUN PARAMETERS			
		aflowavel= 1,000,000 Base Case	aflowavel= 1,000,000 vcargo = 0	aflowavel= 6,000,000 Base Case	aflowavel= 6,000,000 cfixstor=48
500	RFR at Specified L-D rate, D&S mode	$ 6.85 @2000	$ 6.71 @2000	$ 3.87 @6000	$ 3.87 @6000
	RFR at Specified L-D rate, Integral mode	13.84 "	13.69 "	7.71 "	7.94 "
	RFR at Specified L-D rate, D&S mode	5.49 @12000	5.32 @12000	2.55 @54000	2.55 @54000
	RFR at Specified L-D rate, Integral mode	4.65 "	4.47 "	2.17 "	2.30 "
	RFR at L-D Rate Tradeoff Point	About $5.55	About $5.35	About $2.68	About $2.65
	D&S/Integral mode L-D rate Tradeoff Point (LT/day)	Between 8000 & 10000	Between 8000 & 10000	Between 30000 & 38000	Between 30000 & 38000
	% Port Time is of Total Voyage Time at L-D rate Tradeoff Point	72-59	59-37	45-39	45-39
2000	RFR at Specified L-D rate, D&S mode	$14.85 @2000	$14.31 @2000	$ 8.35 @6000	$ 8.35 @6000
	RFR at Specified L-D rate, Integral mode	21.48 "	21.23 "	16.03 "	16.45 "
	RFR at Specified L-D rate, D&S mode	10.46 @12000	9.56 @12000	6.17 @54000	6.17 @54000
	RFR at Specified L-D rate, Integral mode	8.79 "	8.27 "	5.84 "	6.03 "
	RFR at L-D Rate Tradeoff Point	11.67	11.13	6.14	6.15
	D&S/Integral mode L-D rate Tradeoff Point (LT/day)	Near 6,000	Near 6,000	Near 38,000	Between 38,000 & 46000
	% Port Time is of Total Voyage Time at L-D rate Tradeoff Point	49	49	23	23-19
3500	RFR at Specified L-D rate, D&S mode	$19.91 @2000	$19.16 @2000	$12.08 @6000	$12.08 @6000
	RFR at Specified L-D rate, Integral mode	26.82 "	25.82 "	24.28 "	24.84 "
	RFR at Specified L-D rate, D&S mode	15.03 @12000	14.10 @12000	9.85 @54000	9.85 @54000
	RFR at Specified L-D rate, Integral mode	13.25 "	12.24 "	10.08 "	10.26 "
	RFR at L-D Rate Tradeoff Point	15.85	about $14.50	---	---
	D&S/Integral mode L-D rate Tradeoff Point (LT/day)	Between 4000 & 6000	Between 6000 & 8000	None: D&S mode always favorable	None: D&S mode always favorable
	% Port Time is of Total Voyage Time at L-D rate Tradeoff Point	48	48-36	---	---
5000	RFR at Specified L-D rate, D&S mode	$24.56 @2000	$23.26 @2000	$15.31 @6000	$15.31 @6000
	RFR at Specified L-D rate, Integral mode	31.42 "	29.99 "	37.28 "	37.79 "
	RFR at Specified L-D rate, D&S mode	18.55 @12000	17.31 @12000	12.97 @54000	12.97 @54000
	RFR at Specified L-D rate, Integral mode	16.07 "	15.50 "	15.43 "	15.61 "
	RFR at L-D Rate Tradeoff Point	About $19.25	About $17.95	---	---
	D&S/Integral mode L-D rate Tradeoff Point (LT/day)	Between 6000 & 8000	Between 6000 & 8000	None: D&S mode always favorable	None: D&S mode always favorable
	% Port Time is of Total Voyage Time at L-D rate Tradeoff Point	36-32	32- 24	---	---

TABLE 5.4...(Continued)

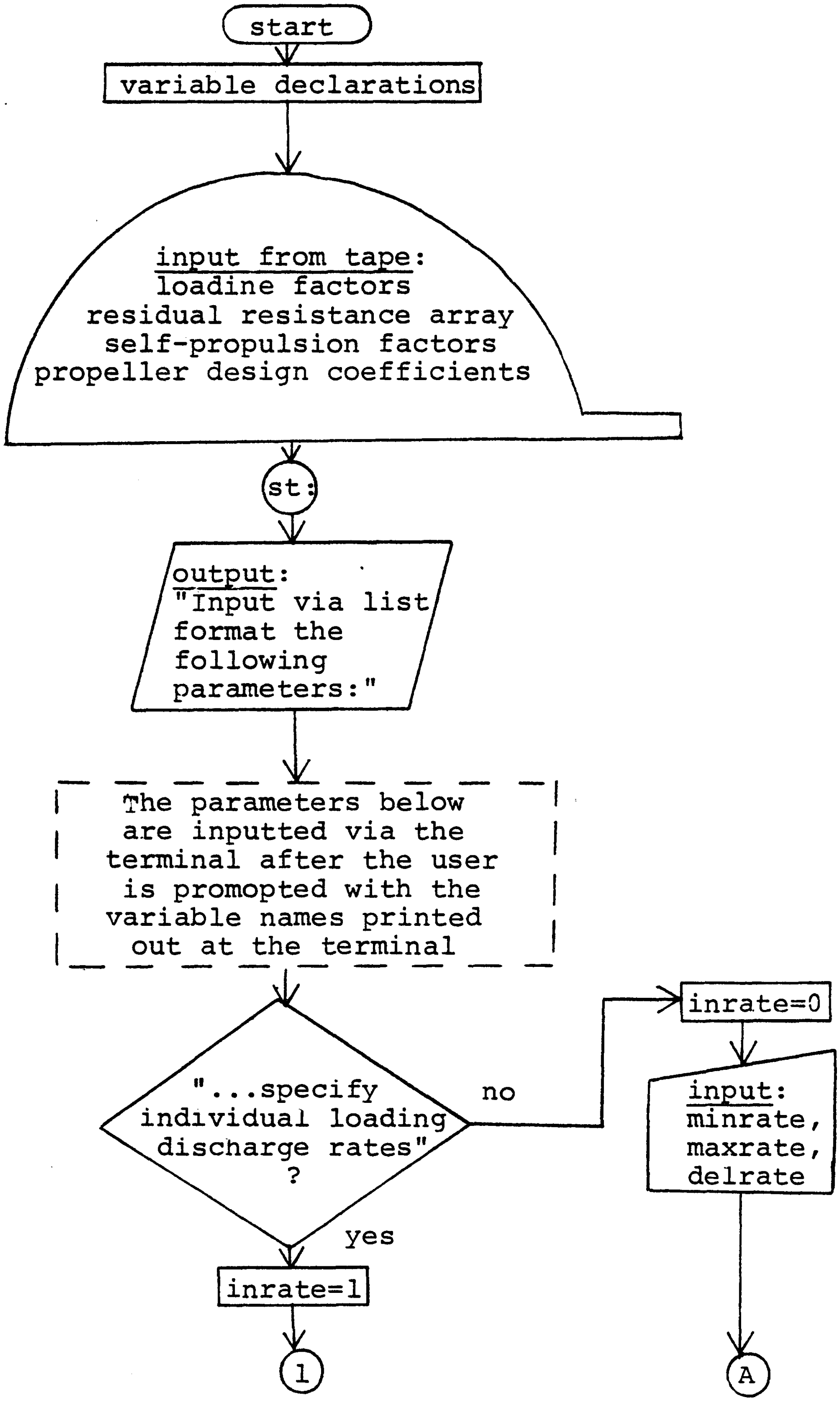
TABULAR SUMMARY FOR SENSITIVITY

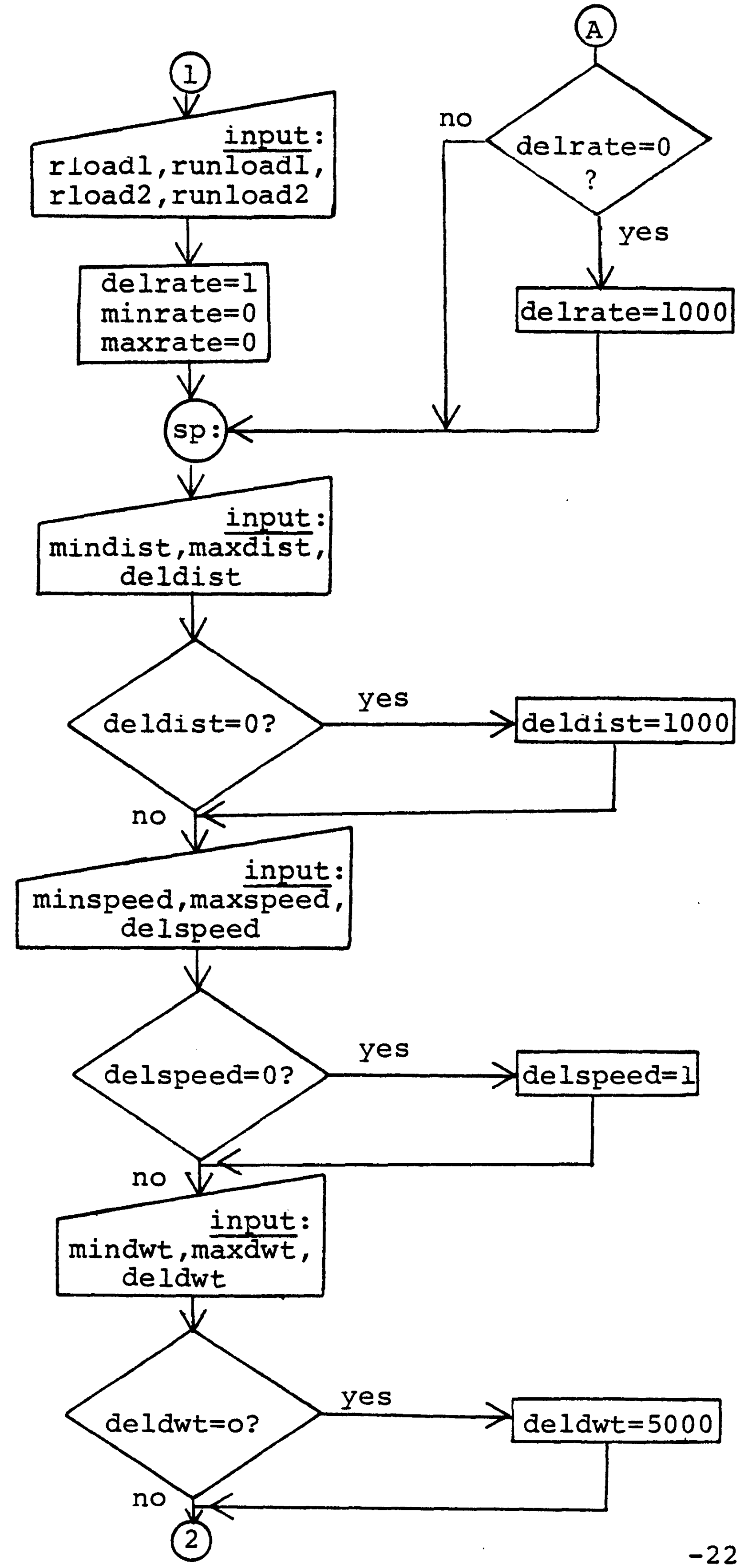
Port Separation Distance (Nautical Miles)	OUTPUT PARAMETERS	SENSITIVITY RUN PARAMETERS			
		aflowavel= 1,000,000 cfixstor=48	aflowavel= 1,000,000 cfixstor=48 cvastor = 1	aflowavel= 1,000,000 cfixstor=48 maxtl=38	aflowavel= 1,000,000 maxtl=38
500	RFR at Specified L-D rate, D&S mode	$ 6.85 @2000	$ 5.65 @3000	$ 5.65 @3000	$ 6.85 @2000
	RFR at Specified L-D rate, Integral mode	14.18 "	11.61 "	10.61 "	13.84 "
	RFR at Specified L-D rate, D&S mode	5.49 @12000	5.49 @12000	5.52 @12000	5.52 @12000
	RFR at Specified L-D rate, Integral mode	5.04 "	6.09 "	5.36	5.07 "
	RFR at L-D Rate Tradeoff Point	5.48	---	About $5.50	About $5.55
	D&S/Integral mode L-D rate Tradeoff Point (LT/day)	Near 10000	None: D&S Mode always favorable	Near 12000	Between 10000 & 12000
	% Port Time is of Total Voyage Time at L-D rate Tradeoff Point	57	---	48	48-29
2000	RFR at Specified L-D rate, D&S mode	$14.85 @2000	$12.50 @3000	$14.43 @3000	$16.05 @2000
	RFR at Specified L-D rate, Integral mode	22.31 "	18.85 "	19.47 "	22.84 "
	RFR at Specified L-D rate, D&S mode	10.46 @12000	10.46 @12000	12.84 @12000	12.84 @12000
	RFR at Specified L-D rate, Integral mode	9.87	11.05 "	12.39 "	12.05 "
	RFR at L-D Rate Tradeoff Point	About $10.70	About $10.00	About $12.80	About $$13.45
	D&S/Integral mode L-D rate Tradeoff Point (LT/day)	Between 10000 & 12000	Between 18000 & 21000	Between 9000 & 12000	Between 6000 & 8000
	% Port Time is of Total Voyage Time at L-D rate Tradeoff Point	48-29	32-27	32-24	37-32
3500	RFR at Specified L-D rate, D&S mode	$19.91 @2000	$16.34 @3000	$20.11 @3000	$22.90 @2000
	RFR at Specified L-D rate, Integral mode	27.95 "	23.01 "	26.72 "	30.89 "
	RFR at Specified L-D rate, D&S mode	15.03 @12000	15.03 @12000	18.85 @12000	18.85 @12000
	RFR at Specified L-D rate, Integral mode	14.04 "	15.04 "	18.79 "	18.29 "
	RFR at L-D Rate Tradeoff Point	About $5.30	About 15.04	About 19.00	About $20,000
	D&S/Integral mode L-D rate Tradeoff Point (LT/day)	Near 8000	12000	Between 6000 & 12000	Between 6000 & 8000
	% Port Time is of Total Voyage Time at L-D rate Tradeoff Point	36	24	24-17	26-24
5000	RFR at Specified L-D rate, D&S mode	$24.56 @2000	$21.24 @3000	$25.10 @3000	$29.72 @2000
	RFR at Specified L-D rate, Integral mode	33.00 "	29.35 "	32.87 "	39.44 "
	RFR at Specified L-D rate, D&S mode	18.55 @12000	18.55 @12000	24.25 @12000	24.25 @12000
	RFR at Specified L-D rate, Integral mode	17.20 "	20.20 "	24.28 "	23.72 "
	RFR at L-D Rate Tradeoff Point	About $19.00	About $18.05	About $24.25	About $24.60
	D&S/Integral mode L-D rate Tradeoff Point (LT/day)	Near 8000	Near 15000	12000	Between 8000 & 10000
	% Port Time is of Total Voyage Time at L-D rate Tradeoff Point	36	19	17	25-19

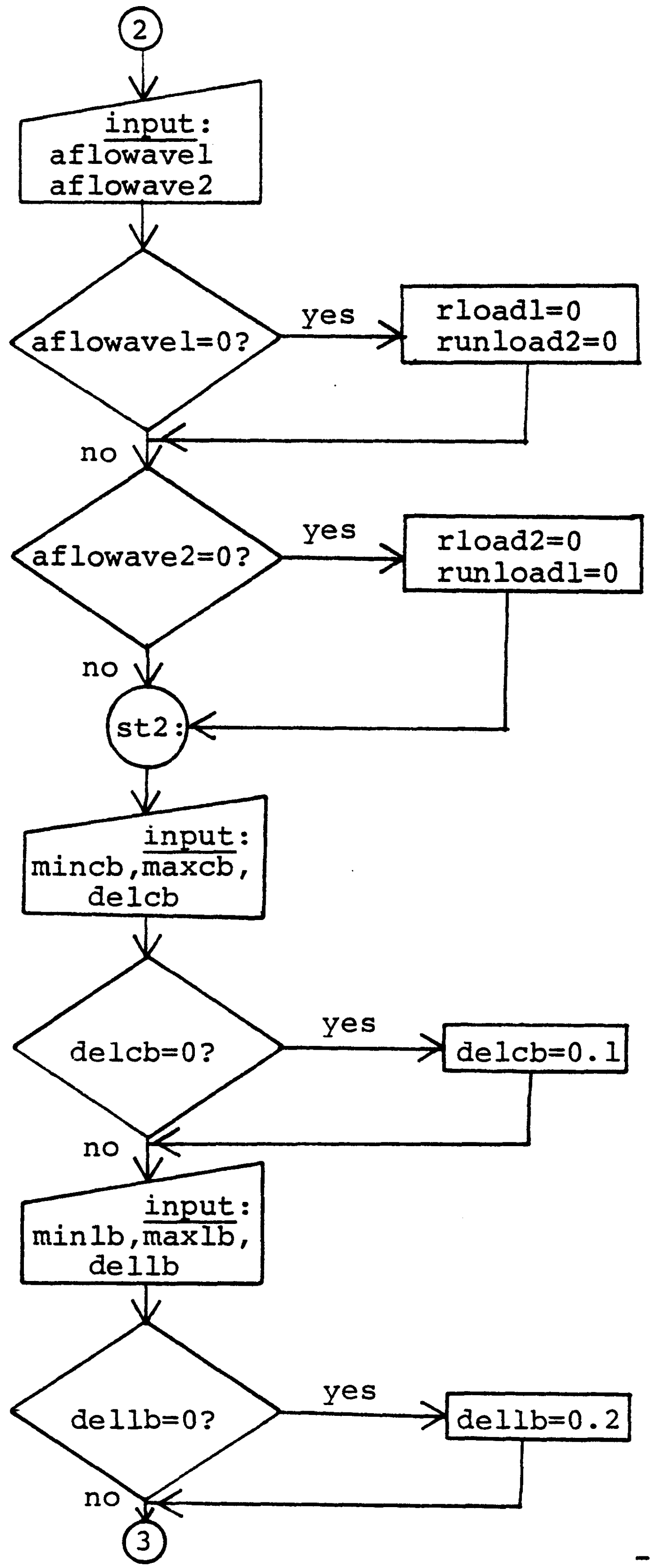
APPENDIX A

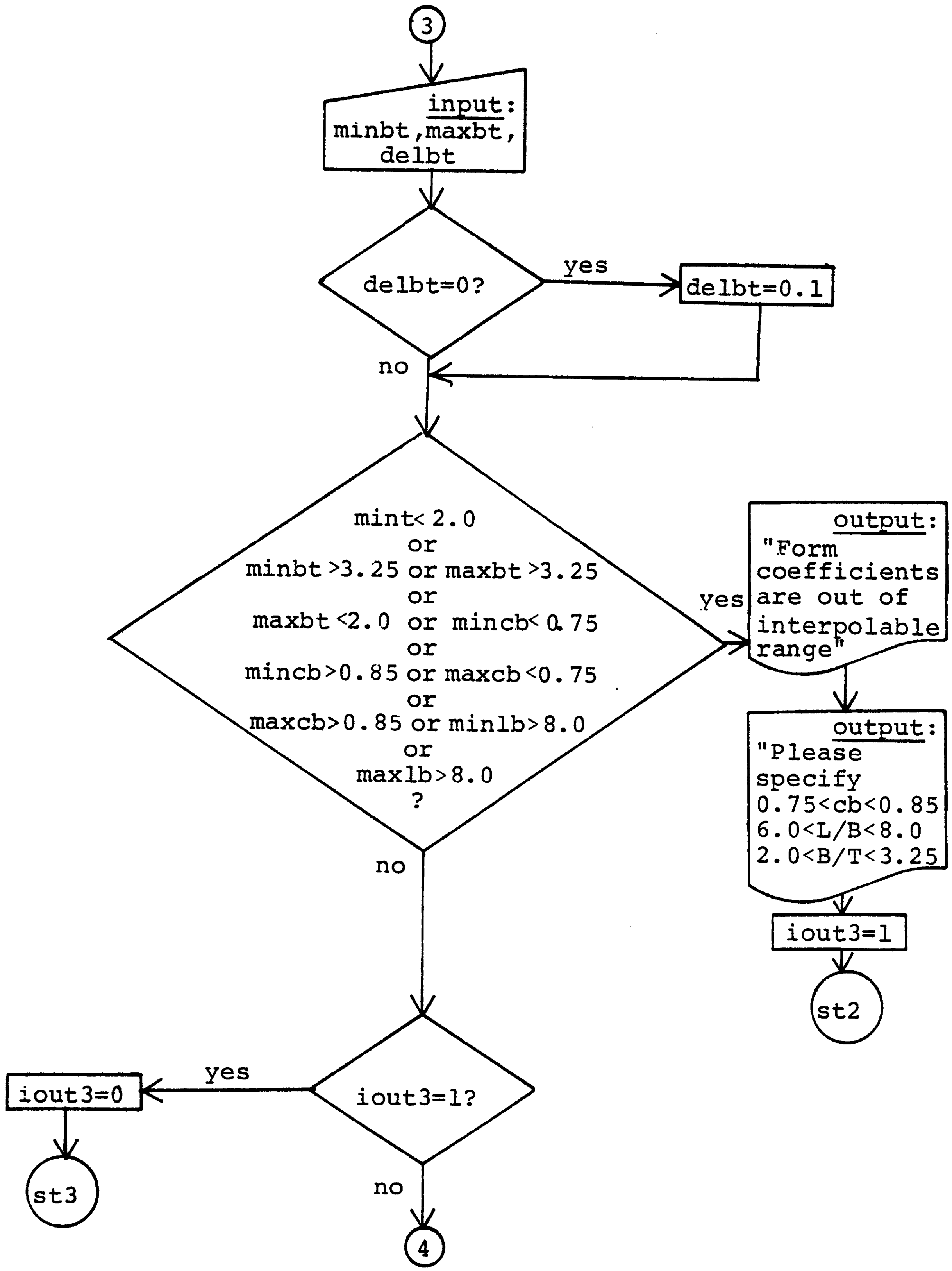
DOCUMENTATION FOR THE DROP-AND-SWAP COMPUTER MODEL

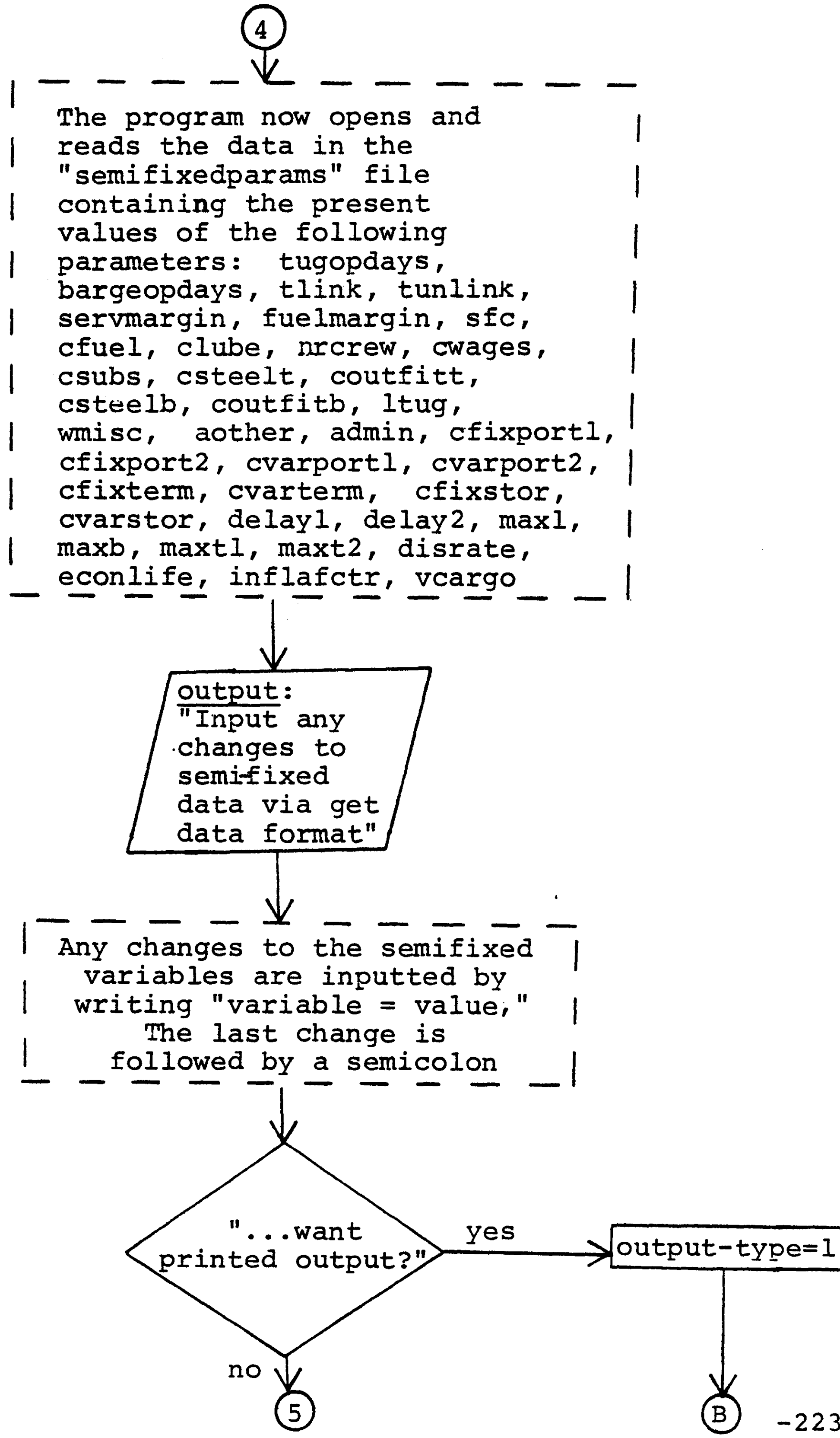
A.1 FLOWCHART OF THE DROP-AND-SWAP COMPUTER MODEL

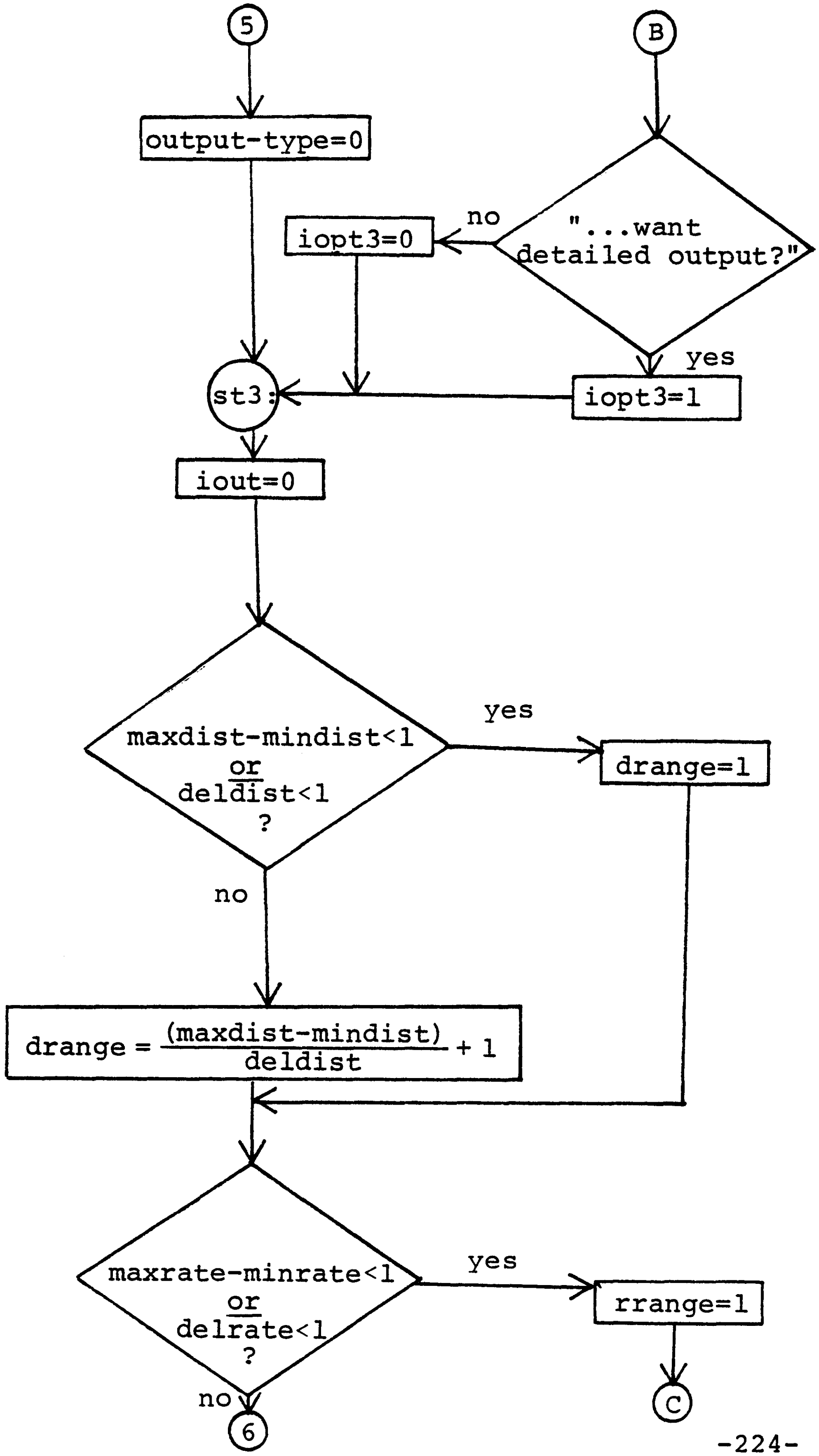


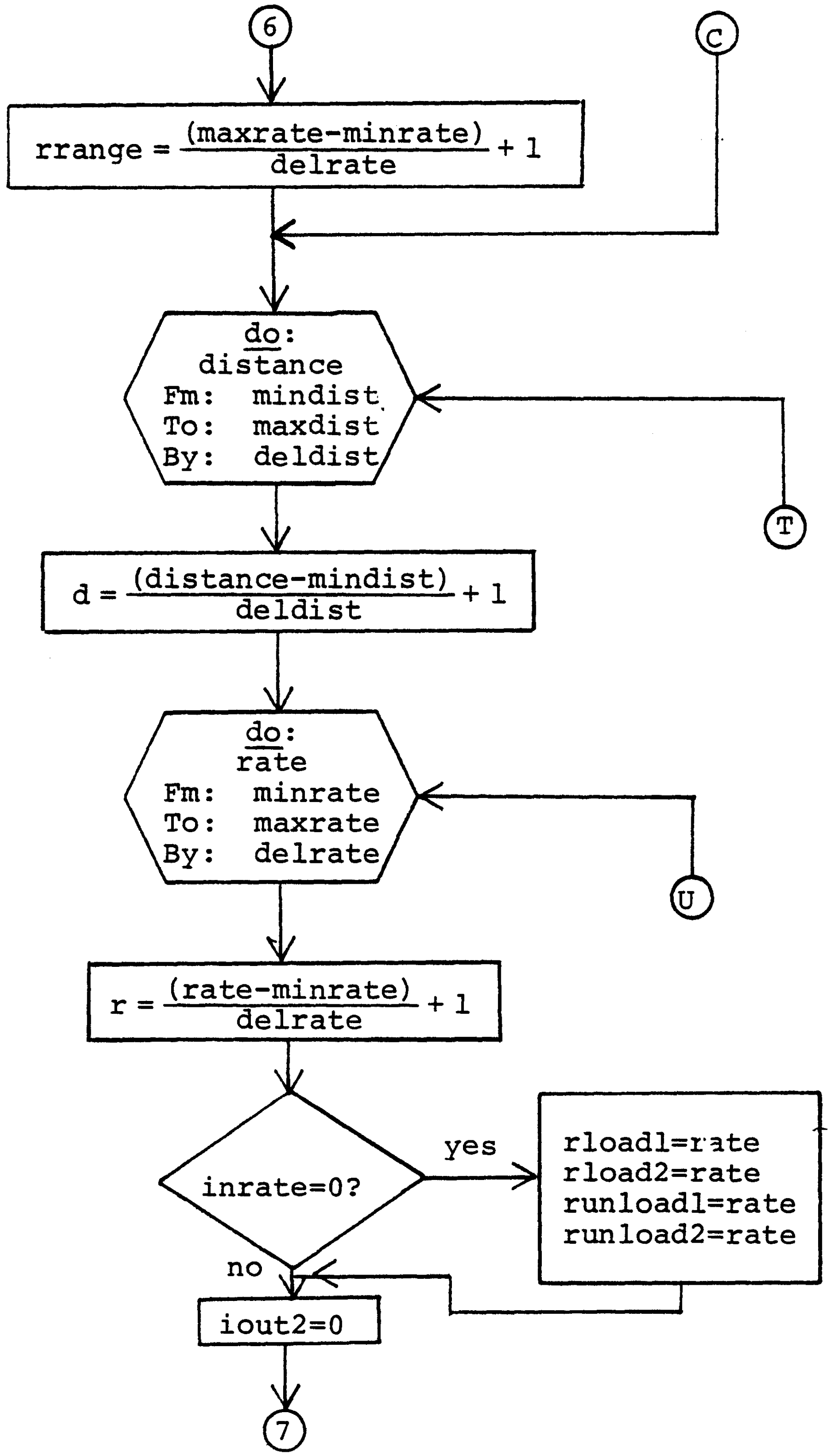

A
1
input:
rload1,runload1,
rload2,runload2
delrate=1
minrate=0
maxrate=0
no
delrate=0
?
yes
delrate=1000
sp:
input:
mindist,maxdist,
deldist
deldist=0?
yes
deldist=1000
no
input:
minspeed,maxspeed,
delspeed
delspeed=0?
yes
delspeed=1
no
input:
mindwt,maxdwt,
deldwt
deldwt=o?
yes
deldwt=5000
no
2

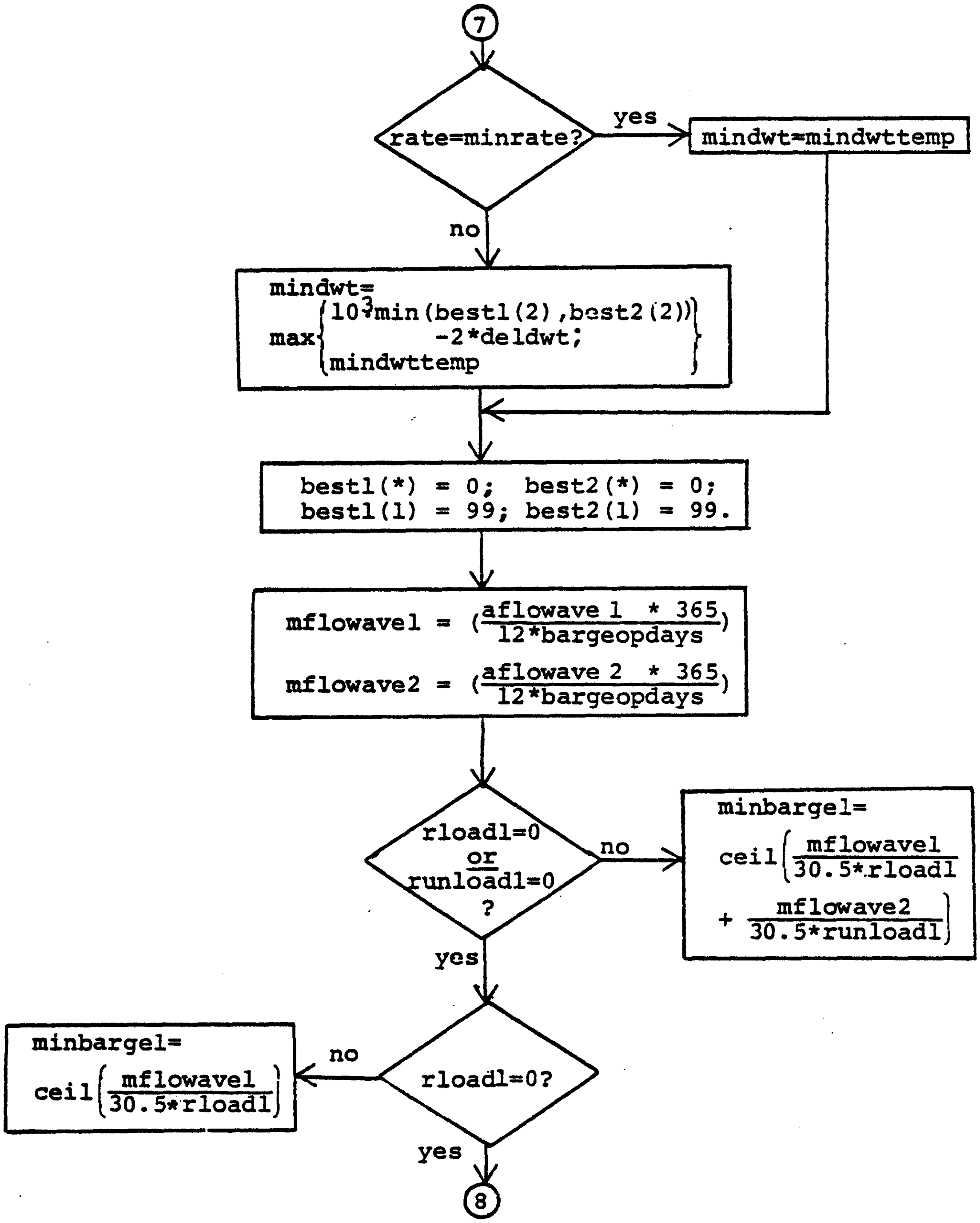

2
input:
aflowavel
aflowave2
aflowavel=0?
yes
rloadl=0
runload2=0
no
aflowave2=0?
yes
rload2=0
runloadl=0
no
st2:
input:
mincb,maxcb,
delcb
delcb=0?
yes
delcb=0.1
no
input:
minlb,maxlb,
dellb
dellb=0?
yes
dellb=0.2
no
3

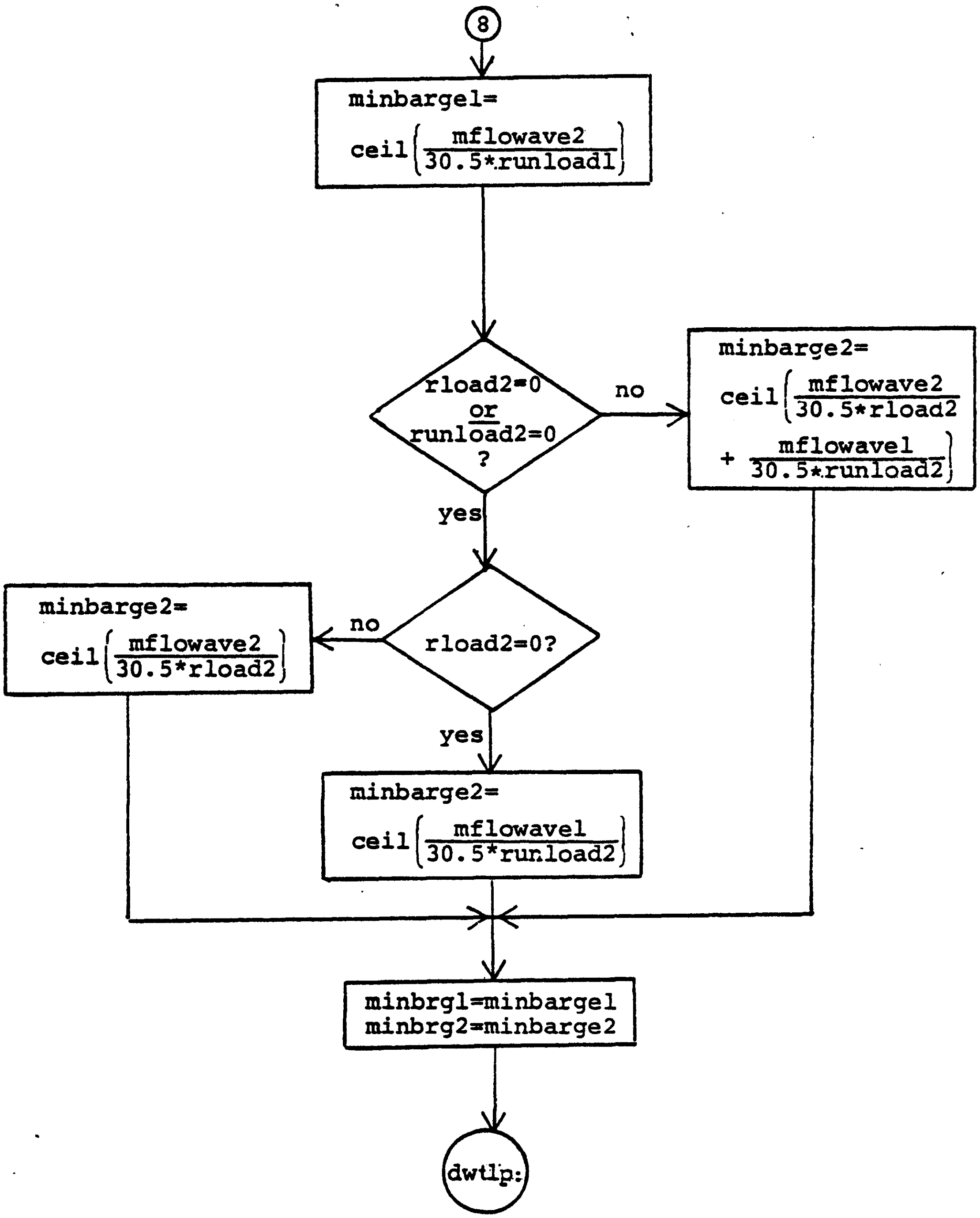

3
input:
minbt,maxbt,
delbt
delbt=0?
yes
delbt=0.1
no
mint< 2.0
or
minbt >3.25 or maxbt >3.25
or
maxbt <2.0 or mincb< 0.75
or
mincb>0.85 or maxcb <0.75
or
maxcb>0.85 or minlb>8.0
or
maxlb>8.0
?
yes
output:
"Form
coefficients
are out of
interpolable
range"
output:
"Please
specify
0.75<cb<0.85
6.0<L/B<8.0
2.0<B/T<3.25
iout3=1
st2
no
iout3=1?
yes
iout3=0
st3
no
4

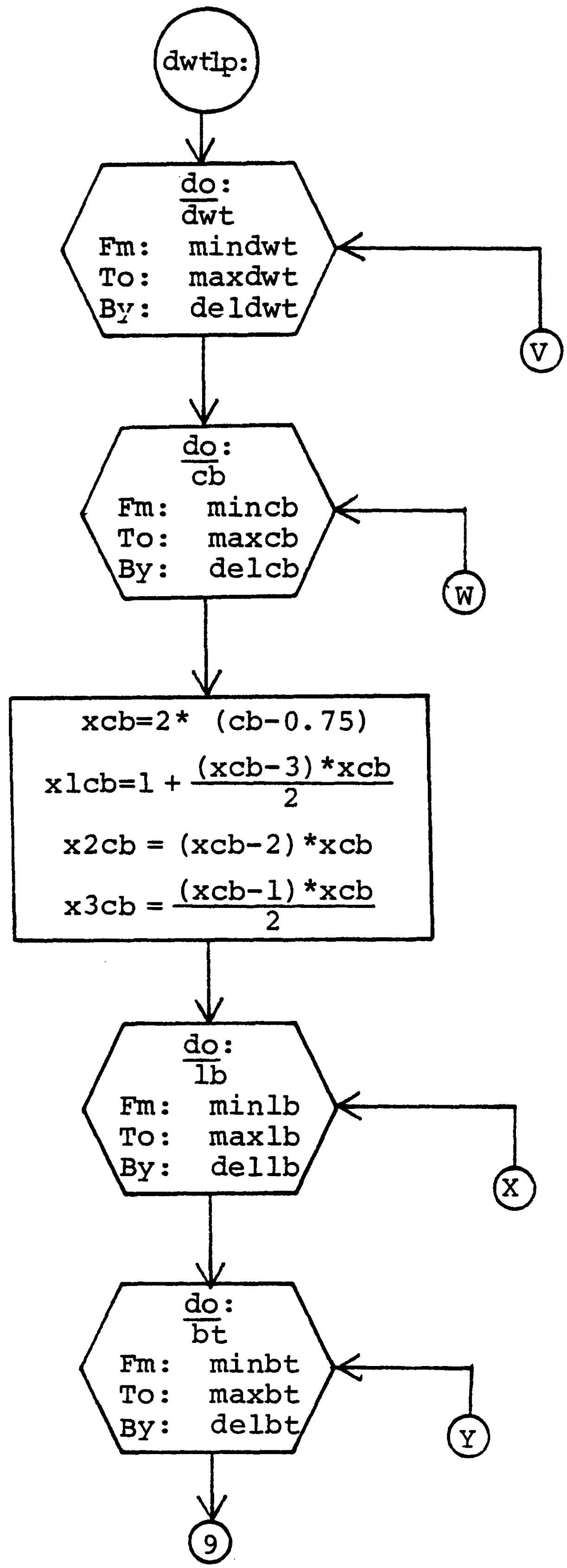

4
The program now opens and reads the data in the "semifixedparams" file containing the present values of the following parameters: tugopdays, bargeopdays, tlink, tunlink, servmargin, fuelmargin, sfc, cfuel, clube, nrcrew, cwages, csubs, csteelt, coutfitt, csteelb, coutfitb, ltug, wmisc, aother, admin, cfixportl, cfixport2, cvarportl, cvarport2, cfixterm, cvarterm, cfixstor, cvarstor, delay1, delay2, maxl, maxb, maxtl, maxt2, disrate, econlife, inflafctr, vcargo
output: "Input any changes to semi-fixed data via get data format"
Any changes to the semifixed variables are inputted by writing "variable = value," The last change is followed by a semicolon
"...want printed output?"
yes
output-type=1
no
5
B


5
B
output-type=0
"...want detailed output?"
no
iopt3=0
yes
iopt3=1
st3
iout=0
maxdist-mindist<1
or
deldist<1
?
yes
drange=1
no
drange = (maxdist-mindist) / deldist + 1
maxrate-minrate<1
or
delrate<1
?
yes
rrange=1
no
6
C


6
C
rrange = (maxrate-minrate) / delrate + 1
do:
distance
Fm: mindist
To: maxdist
By: deldist
T
d = (distance-mindist) / deldist + 1
do:
rate
Fm: minrate
To: maxrate
By: delrate
U
r = (rate-minrate) / delrate + 1
inrate=0?
yes
rload1=rate
rload2=rate
runload1=rate
runload2=rate
no
iout2=0
7


7
rate=minrate?
yes
mindwt=mindwttemp
no
mindwt=
max{10^3*min(best1(2),best2(2)) -2*deldwt; mindwttemp}
best1(*) = 0; best2(*) = 0;
best1(1) = 99; best2(1) = 99.
mflowave1 = (aflowave 1 * 365 / 12*bargeopdays)
mflowave2 = (aflowave 2 * 365 / 12*bargeopdays)
rload1=0 or runload1=0 ?
no
minbarge1=
ceil(mflowave1 / 30.5*rload1 + mflowave2 / 30.5*runload1)
yes
rload1=0?
no
minbarge1=
ceil(mflowave1 / 30.5*rload1)
yes
8


8
minbarge1=
ceil(mflowave2 / 30.5*runload1)
rload2=0 or runload2=0 ?
no
minbarge2=
ceil(mflowave2 / 30.5*rload2 + mflowave1 / 30.5*runload2)
yes
rload2=0?
no
minbarge2=
ceil(mflowave2 / 30.5*rload2)
yes
minbarge2=
ceil(mflowave1 / 30.5*runload2)
minbrg1=minbarge1
minbrg2=minbarge2
dwtlp:


dwtlp:
do:
dwt
Fm: mindwt
To: maxdwt
By: deldwt
V
do:
cb
Fm: mincb
To: maxcb
By: delcb
W
xcb=2* (cb-0.75)
x1cb=1 + (xcb-3)*xcb / 2
x2cb = (xcb-2)*xcb
x3cb = (xcb-1)*xcb / 2
do:
lb
Fm: minlb
To: maxlb
By: dellb
X
do:
bt
Fm: minbt
To: maxbt
By: delbt
Y
9

9

lbarge75=

$$e^{1.128} * lb^{0.742} * bt^{0.382} * dwt^{0.336}$$

lbarge80=

$$e^{1.108} * lb^{0.739} * bt^{0.381} * dwt^{0.336}$$

lbarge85=

$$e^{1.088} * lb^{0.737} * bt^{0.379} * dwt^{0.336}$$

lbarge=

lbarge75*x1cb
-lbarge80*x2cb
+lbarge85*x3cb

hullwt75=

$$e^{-6.206} * bt^{0.884} * lb^{1.348} * dwt^{1.104}$$

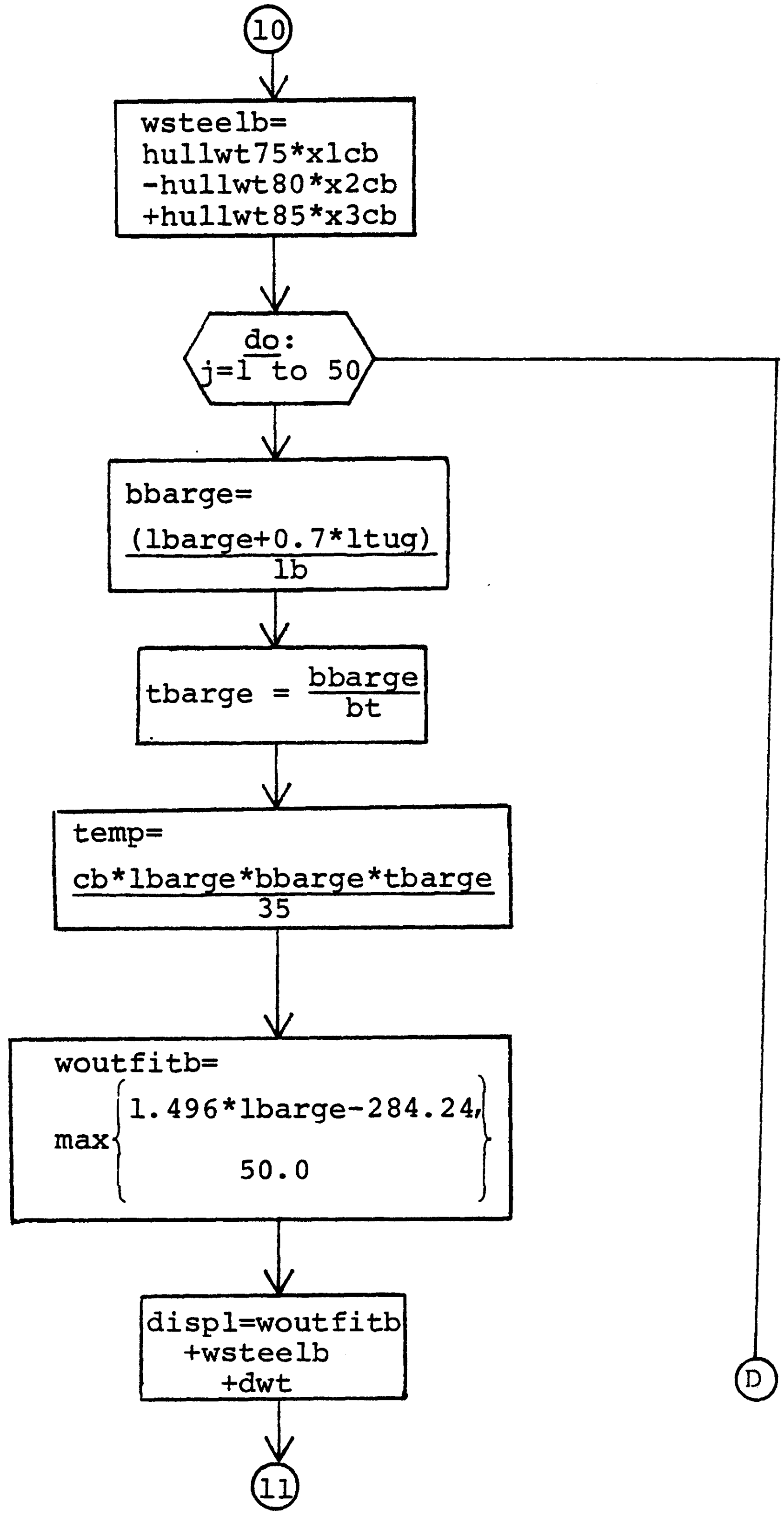
hullwt80=

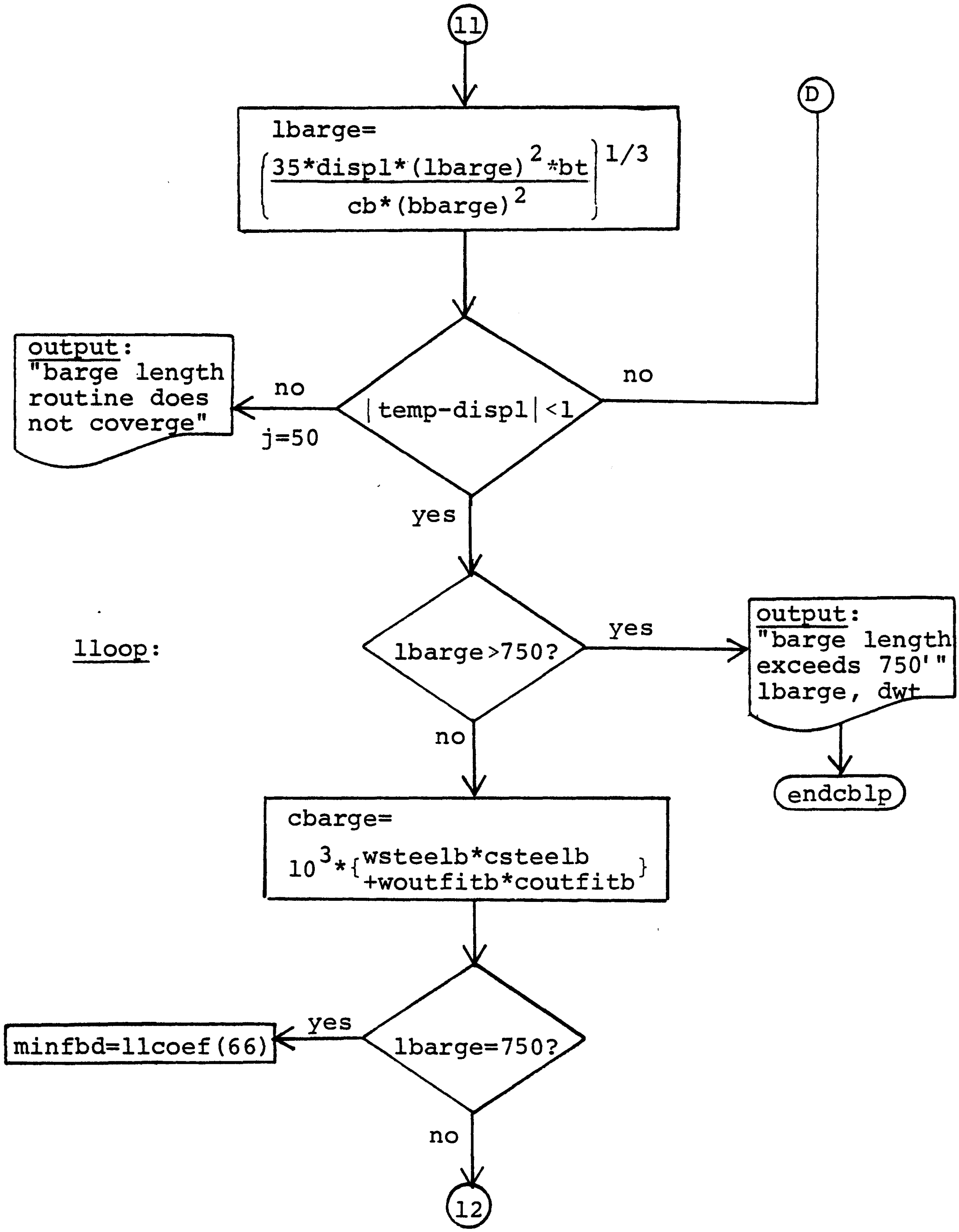
$$e^{-6.396} * bt^{0.909} * lb^{1.368} * dwt^{1.111}$$

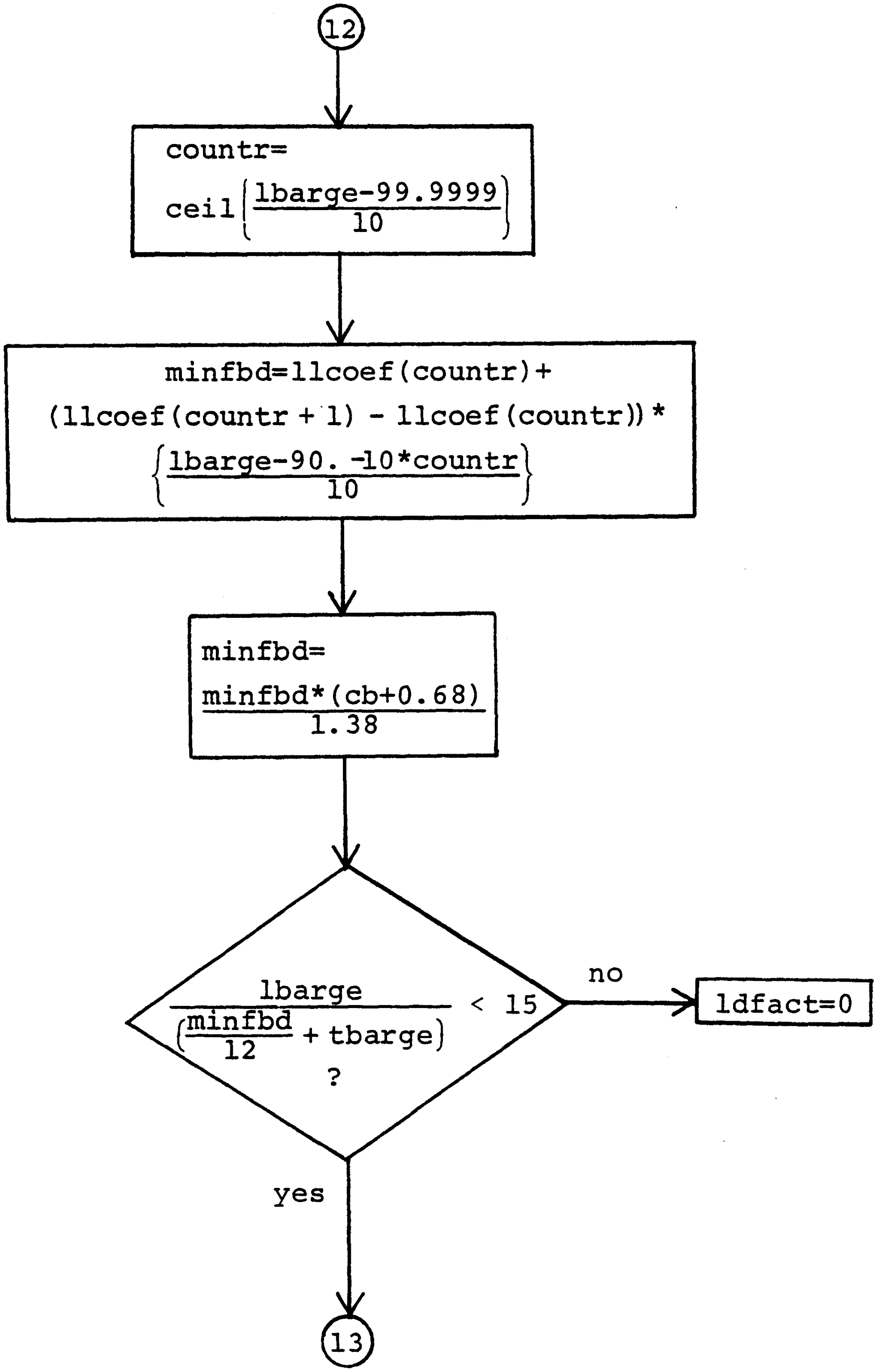
hullwt85=

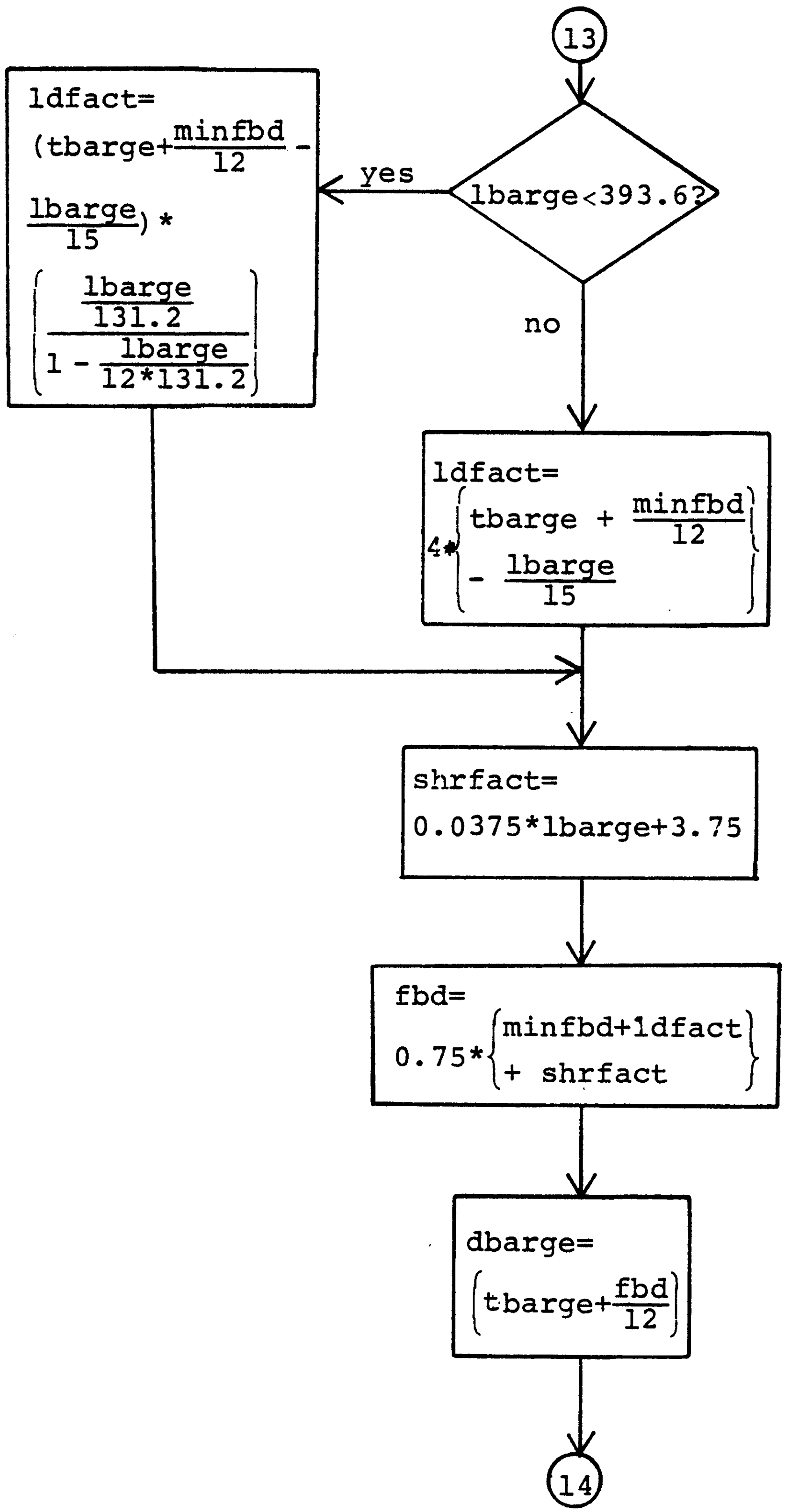
$$e^{-6.569} * bt^{0.930} * lb^{1.391} * dwt^{1.117}$$

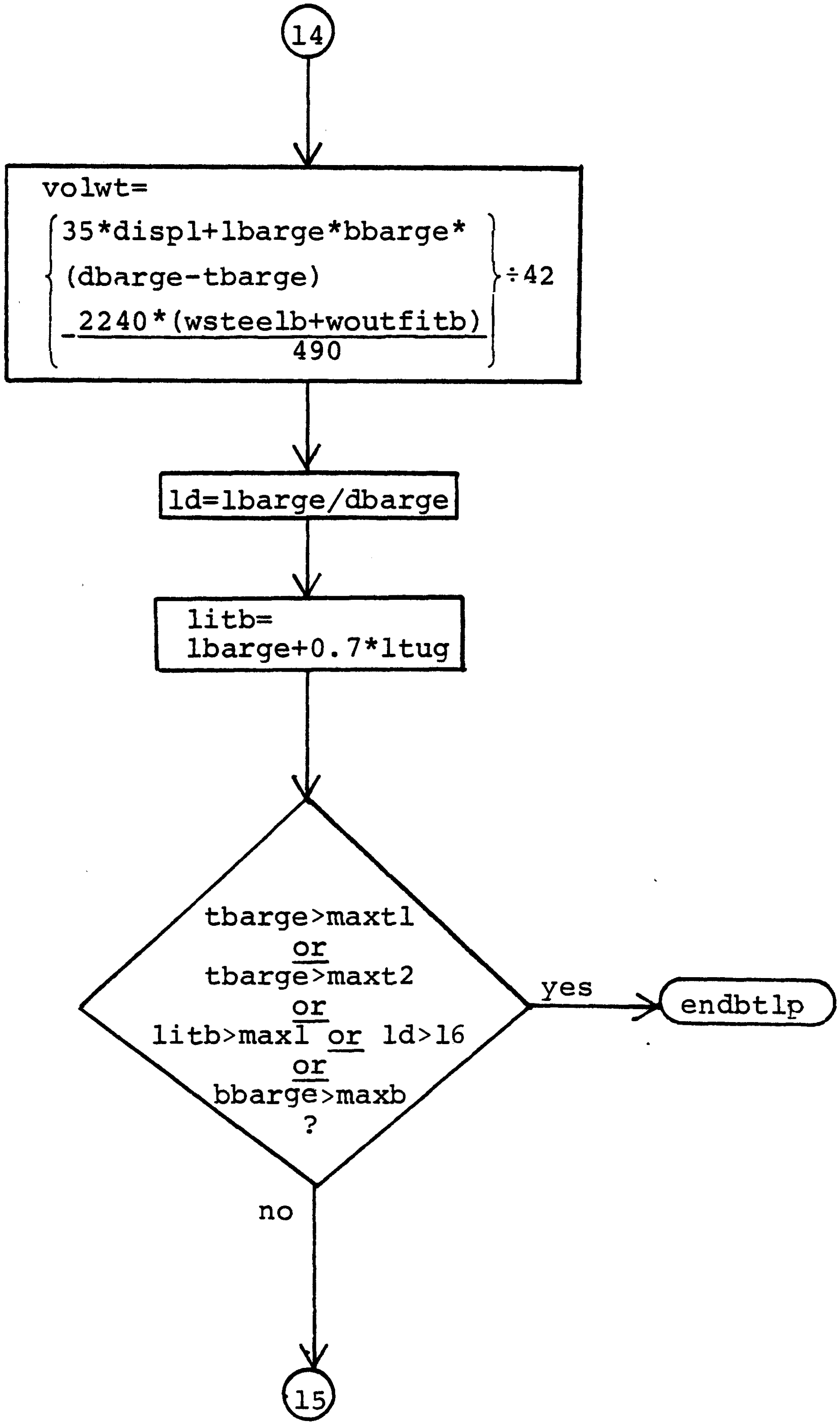
10

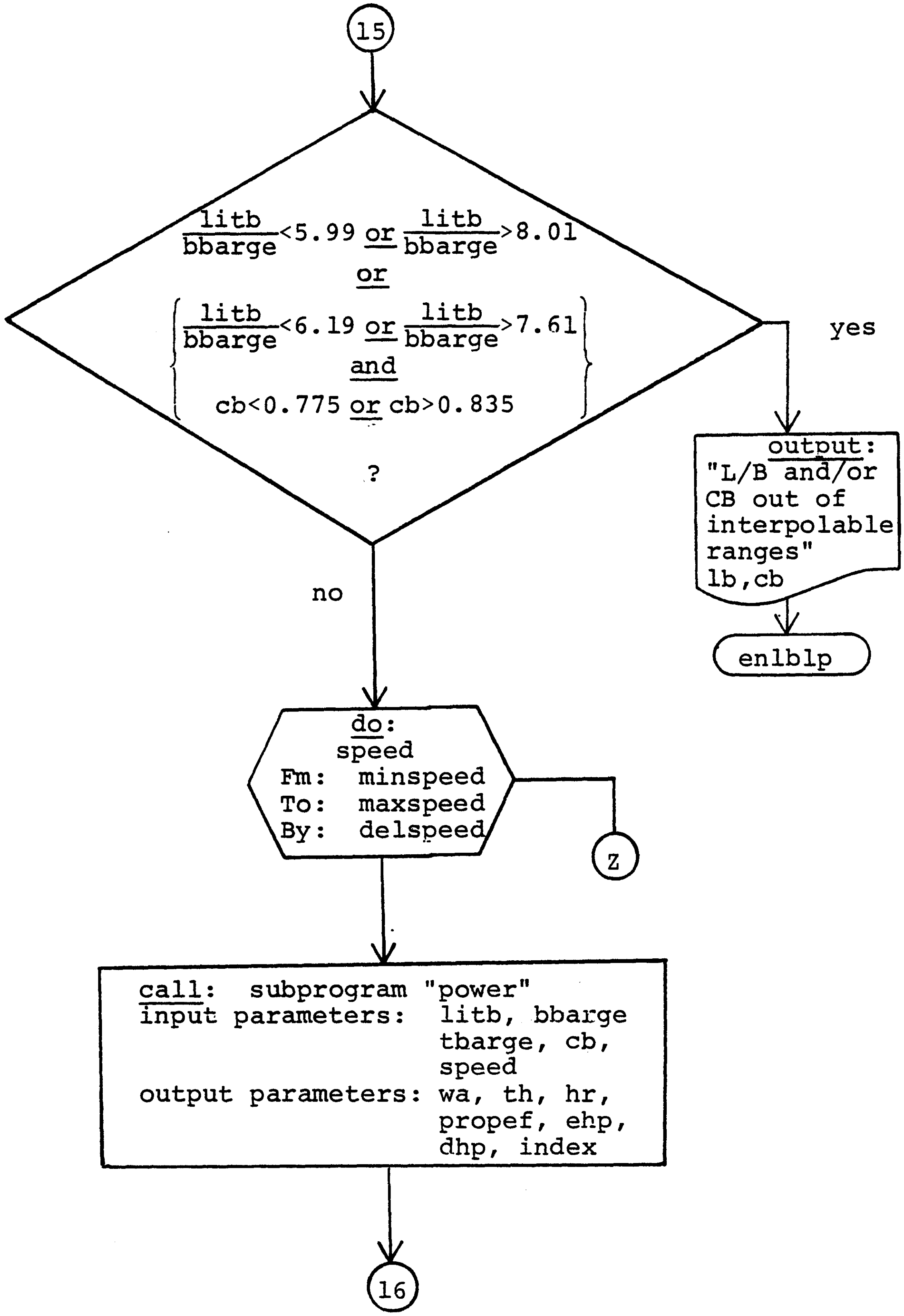

10
wsteelb=
hullwt75*x1cb
-hullwt80*x2cb
+hullwt85*x3cb
do:
j=1 to 50
bbarge=
(lbarge+0.7*ltug)
lb
tbarge = bbarge
bt
temp=
cb*lbarge*bbarge*tbarge
35
woutfitb=
max 1.496*lbarge-284.24,
50.0
displ=woutfitb
+wsteelb
+dwt
D
11

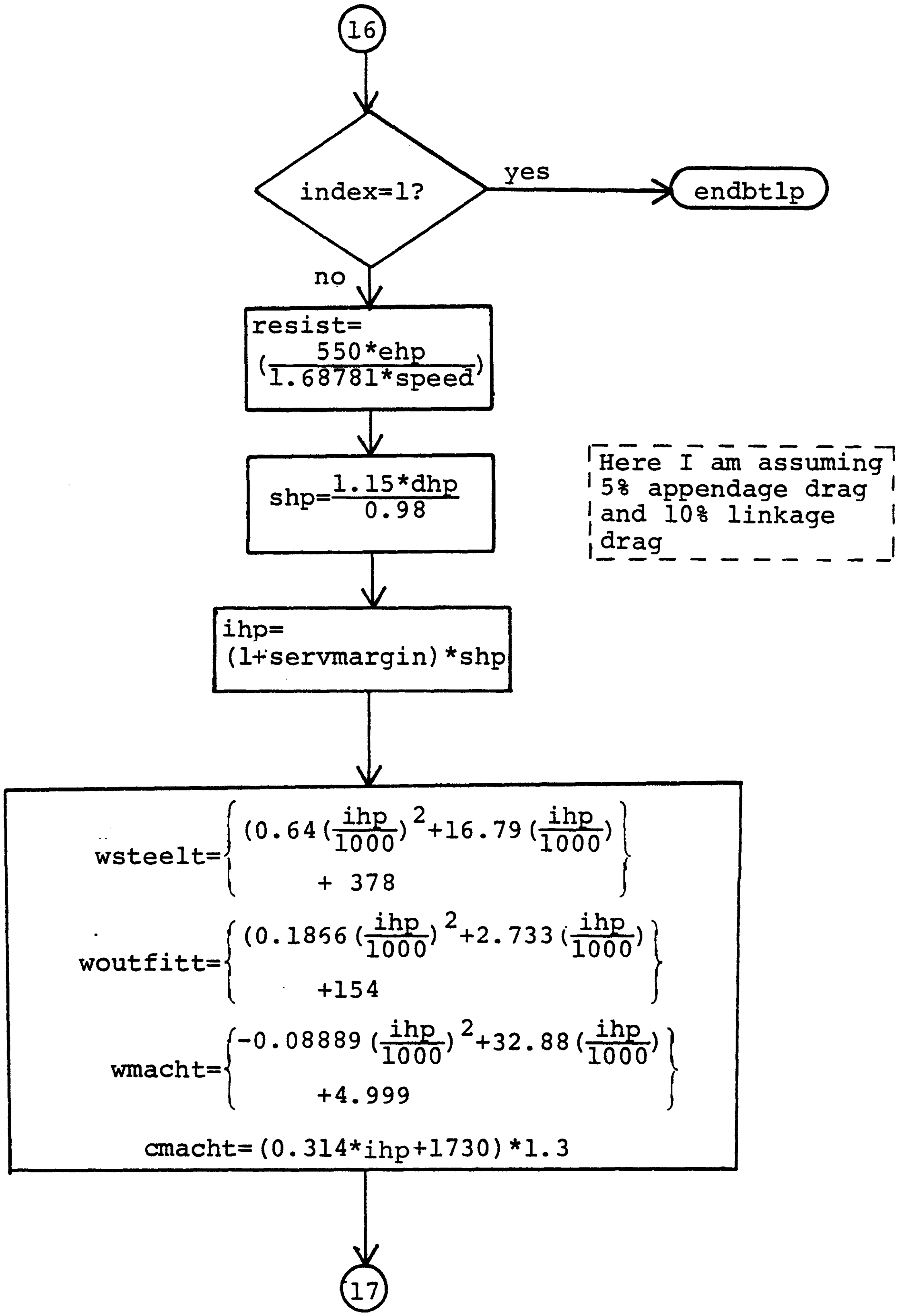

11
lbarge=
{35*displ*(lbarge)^2*bt / cb*(bbarge)^2}^1/3
D
output:
"barge length
routine does
not coverge"
no
j=50
|temp-displ|<1
no
yes
lloop:
lbarge>750?
yes
output:
"barge length
exceeds 750'"
lbarge, dwt
endcblp
no
cbarge=
10^3*{wsteelb*csteelb
+woutfitb*coutfitb}
minfbd=llcoef(66)
yes
lbarge=750?
no
12

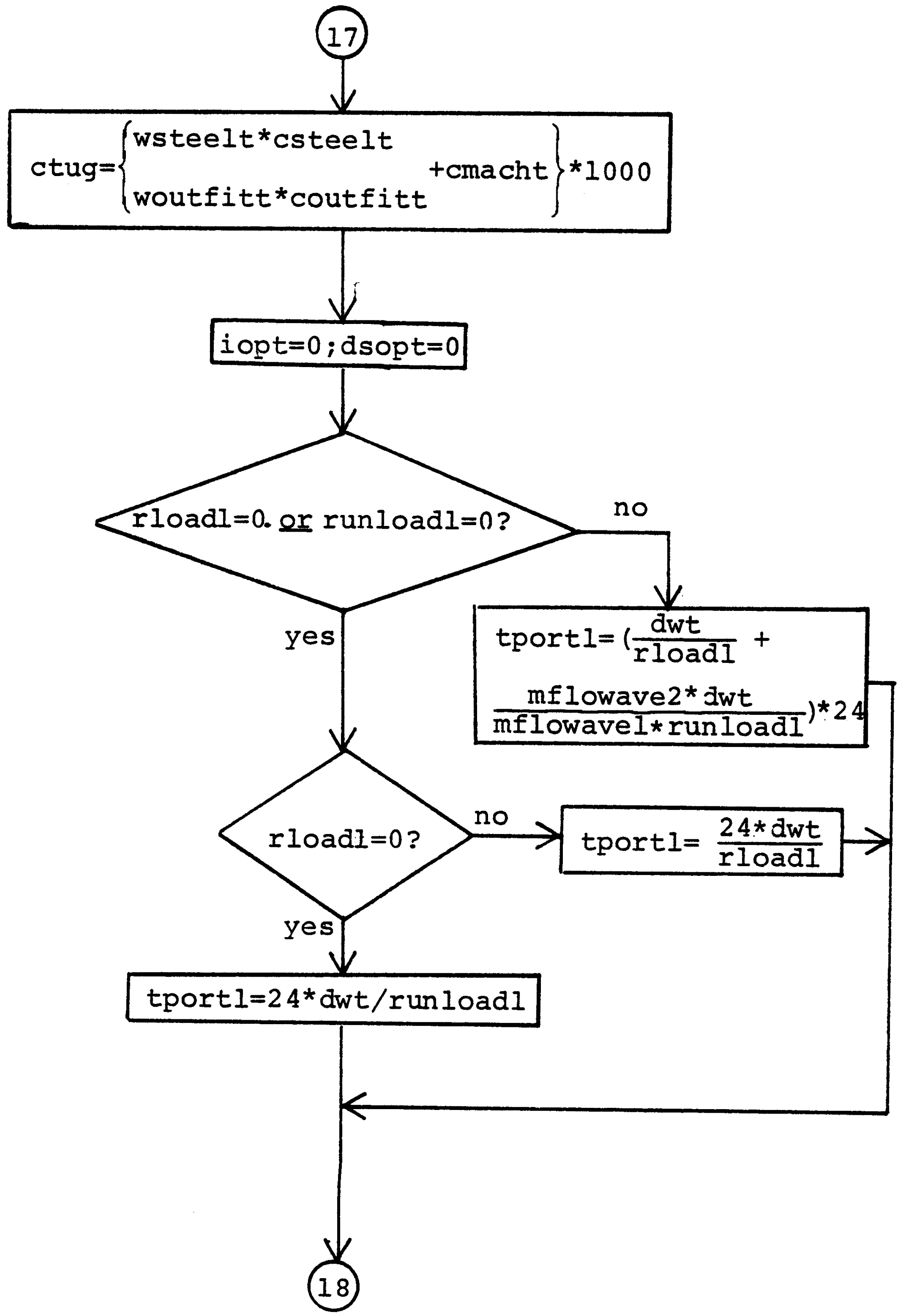

12
countr= ceil{(lbarge-99.9999)/10}
minfbd=llcoef(countr)+ (llcoef(countr + 1) - llcoef(countr))* {(lbarge-90. -10*countr)/10}
minfbd= minfbd*(cb+0.68)/1.38
lbarge/(minfbd/12 + tbarge) < 15 ?
no
ldfact=0
yes
13

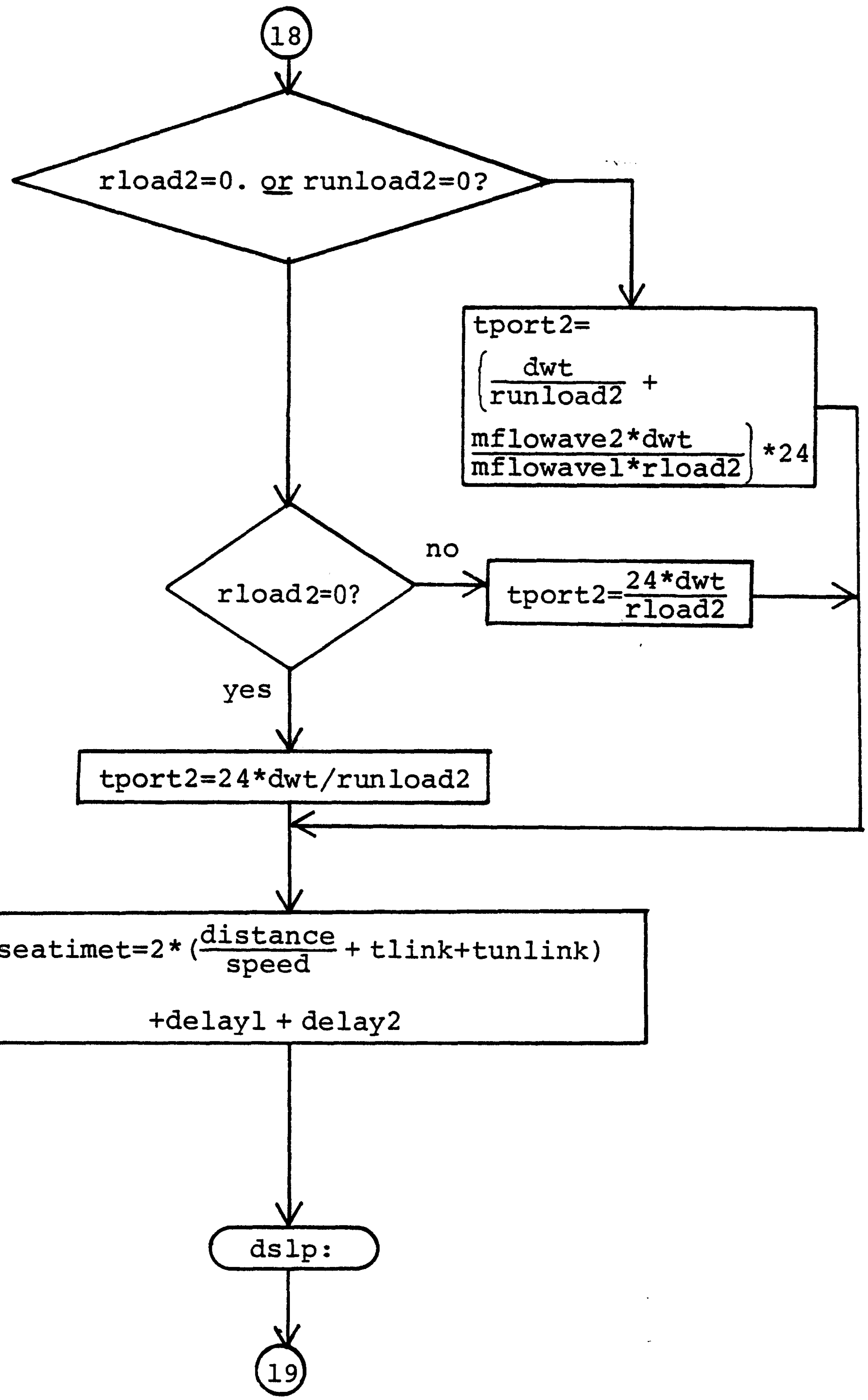

13
lbarge<393.6?
yes
no
ldfact=
(tbarge+minfbd/12 -
lbarge/15) *
{ (lbarge/131.2) / (1 - lbarge/12*131.2) }
ldfact=
4*{ tbarge + minfbd/12
- lbarge/15 }
shrfact=
0.0375*lbarge+3.75
fbd=
0.75*{ minfbd+ldfact
+ shrfact }
dbarge=
{ tbarge+fbd/12 }
14


14
volwt=
35*displ+lbarge*bbarge*
(dbarge-tbarge)
-2240*(wsteelb+woutfitb)
490
÷42
ld=lbarge/dbarge
litb=
lbarge+0.7*ltug
tbarge>maxt1
or
tbarge>maxt2
or
litb>maxl or ld>16
or
bbarge>maxb
?
yes
endbtlp
no
15

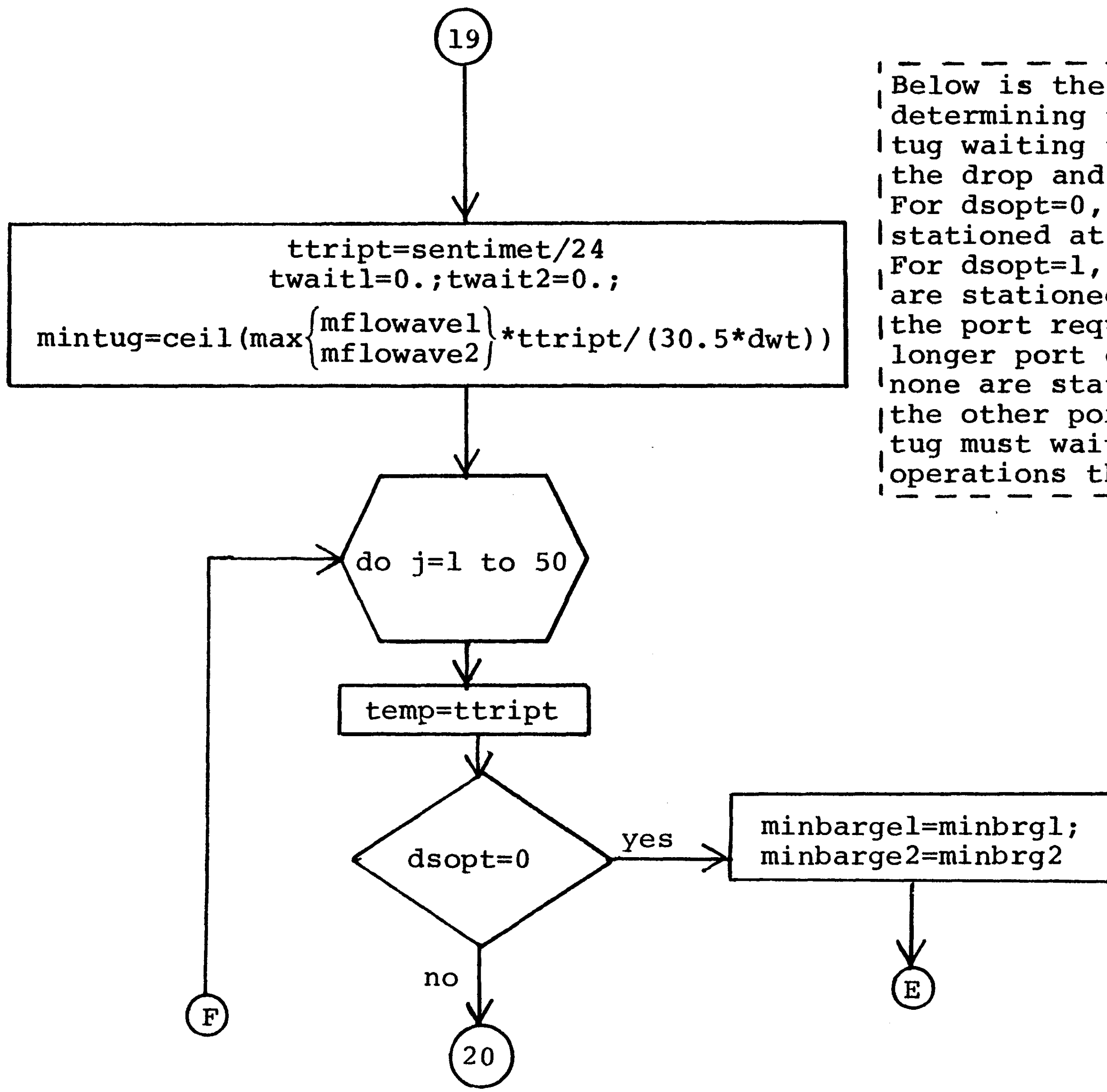

15
litb/bbarge<5.99 or litb/bbarge>8.01
or
litb/bbarge<6.19 or litb/bbarge>7.61
and
cb<0.775 or cb>0.835
?
yes
output:
"L/B and/or CB out of interpolable ranges"
lb,cb
enlblp
no
do:
speed
Fm: minspeed
To: maxspeed
By: delspeed
Z
call: subprogram "power"
input parameters: litb, bbarge tbarge, cb, speed
output parameters: wa, th, hr, propef, ehp, dhp, index
16

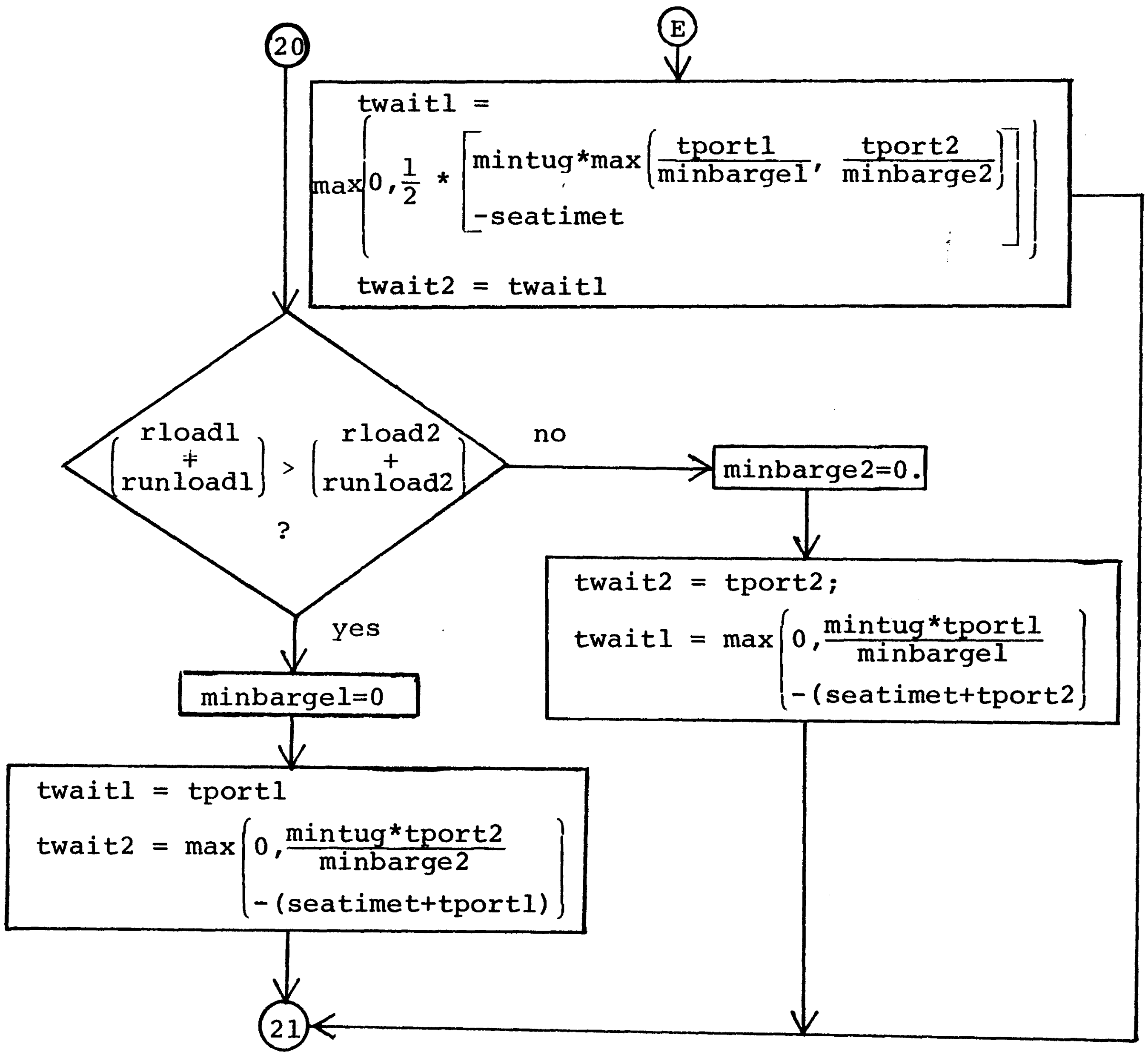

16
index=1?
yes
endbtlp
no
resist= (550*ehp / 1.68781*speed)
shp= 1.15*dhp / 0.98
Here I am assuming 5% appendage drag and 10% linkage drag
ihp= (1+servmargin)*shp
wsteelt={(0.64 (ihp/1000)^2+16.79 (ihp/1000) + 378}
woutfitt={(0.1866 (ihp/1000)^2+2.733 (ihp/1000) +154}
wmacht={-0.08889 (ihp/1000)^2+32.88 (ihp/1000) +4.999}
cmacht=(0.314*ihp+1730)*1.3
17

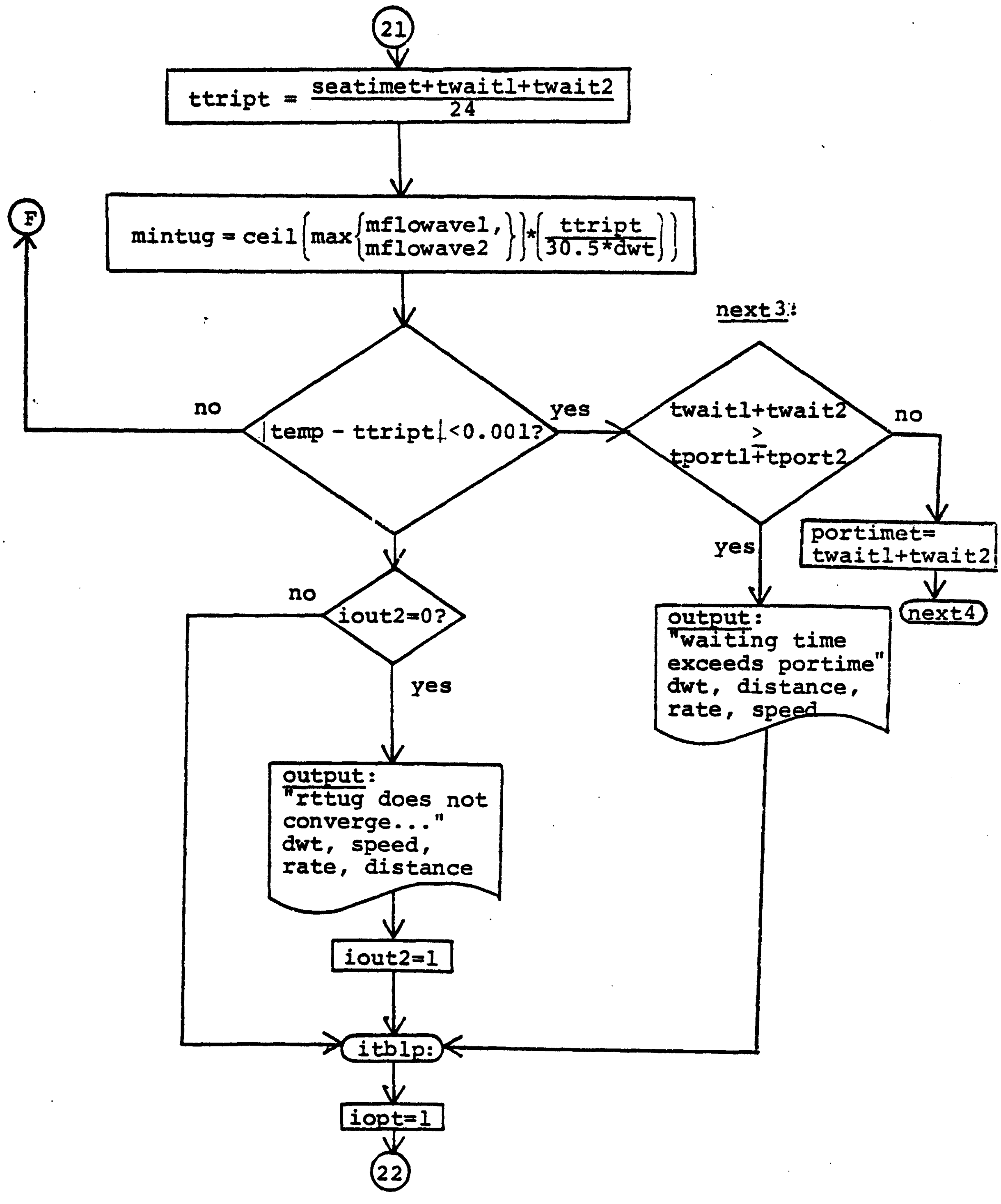

17
ctug={wsteelt*csteelt + woutfitt*coutfitt +cmacht}*1000
iopt=0;dsopt=0
rloadl=0. or runloadl=0?
no
tportl=(dwt/rloadl + mflowave2*dwt/mflowavel*runloadl)*24
yes
rloadl=0?
no
tportl= 24*dwt/rloadl
yes
tportl=24*dwt/runloadl
18


18
rload2=0. or runload2=0?
tport2=
dwt / runload2 +
mflowave2*dwt / mflowave1*rload2 *24
no
rload2=0?
tport2=24*dwt / rload2
yes
tport2=24*dwt/runload2
seatimet=2*(distance / speed + tlink+tunlink)
+delay1 + delay2
dslp:
19

Below is the logic for determining the required tug waiting time for the drop and swap mode. For dsopt=0, barges are stationed at both ports. For dsopt=1, barges are stationed only at the port requiring longer port operations, none are stationed at the other port so the tug must wait for cargo operations there.




20
E
twait1 =
max(0, 1/2 * [mintug*max(tport1/minbarge1, tport2/minbarge2) -seatimet])
twait2 = twait1
(rload1 + runload1) > (rload2 + runload2) ?
no
yes
minbarge2=0.
twait2 = tport2;
twait1 = max(0, mintug*tport1/minbarge1 -(seatimet+tport2))
minbarge1=0
twait1 = tport1
twait2 = max(0, mintug*tport2/minbarge2 -(seatimet+tport1))
21



21
ttript = seatimet+twait1+twait2 / 24
mintug = ceil{max{mflowave1, mflowave2}*{ttript / 30.5*dwt}}
F
no
|temp - ttript|<0.001?
yes
next3:
twait1+twait2 > tport1+tport2
no
yes
portimet= twait1+twait2
next4
output: "waiting time exceeds portime" dwt, distance, rate, speed
no
iout2=0?
yes
output: "rttug does not converge..." dwt, speed, rate, distance
iout2=1
itblp:
iopt=1
22

(22)

$$seatimet = 2*\left(\frac{distance}{speed}\right) + delay1 + delay2$$

portimet = tport1 + tport2

ttript = (seatimet + portimet)/24

$$minitb = ceil\left[max\begin{Bmatrix}mflowave1\\mflowave2\end{Bmatrix} * \left(\frac{ttript}{30.5*dwt}\right)\right]$$

next4:

rseafuel = shp x sfc/2240
rlubeoil = 0.00025 * shp
rportfuel = 0.125

acrew = (cwages + csubs) *nrcrew

fuelcons = (rseafuel*seatimet+reportfuel*portimet
lubecons = rlubeoil*seatimet

nrtrips = tugopdays/ttript
tdwt = dwt+(1+fuelmargin)*fuelcons+wmisc

aflowcap = nrtrips*dwt
afuel = fuelcons*nrtrips*cfuel
alube = lubecons*nrtrips*clube

(23)

23

$$\text{amandr} = \left\{ 128.8 + \frac{4.539*\text{ihp}}{1000} - \left(\frac{0.2*\text{ihp}}{1000}\right)^2 + \frac{2.477*\text{tdwt}}{1000} - 9.107*10^{-3}\left(\frac{\text{tdwt}}{1000}\right)^2 \right\} 1500$$

ainsur=(210+0.0036*ihp+0.0018*tdwt)1500

$\text{asupplies}=(40+0.0018*\text{ihp}+2.6*10^{-4}*\text{tdwt})1300$

aport=(cfixport1+cfixport2+(cvarport1+cvarport2)*dwt)
*nrtrips

acargo=
$\text{vcargo}(\text{aflowave1}+\text{aflowave2})*\text{disrate}*\left(\frac{\text{seatime}}{48*365}\right)$

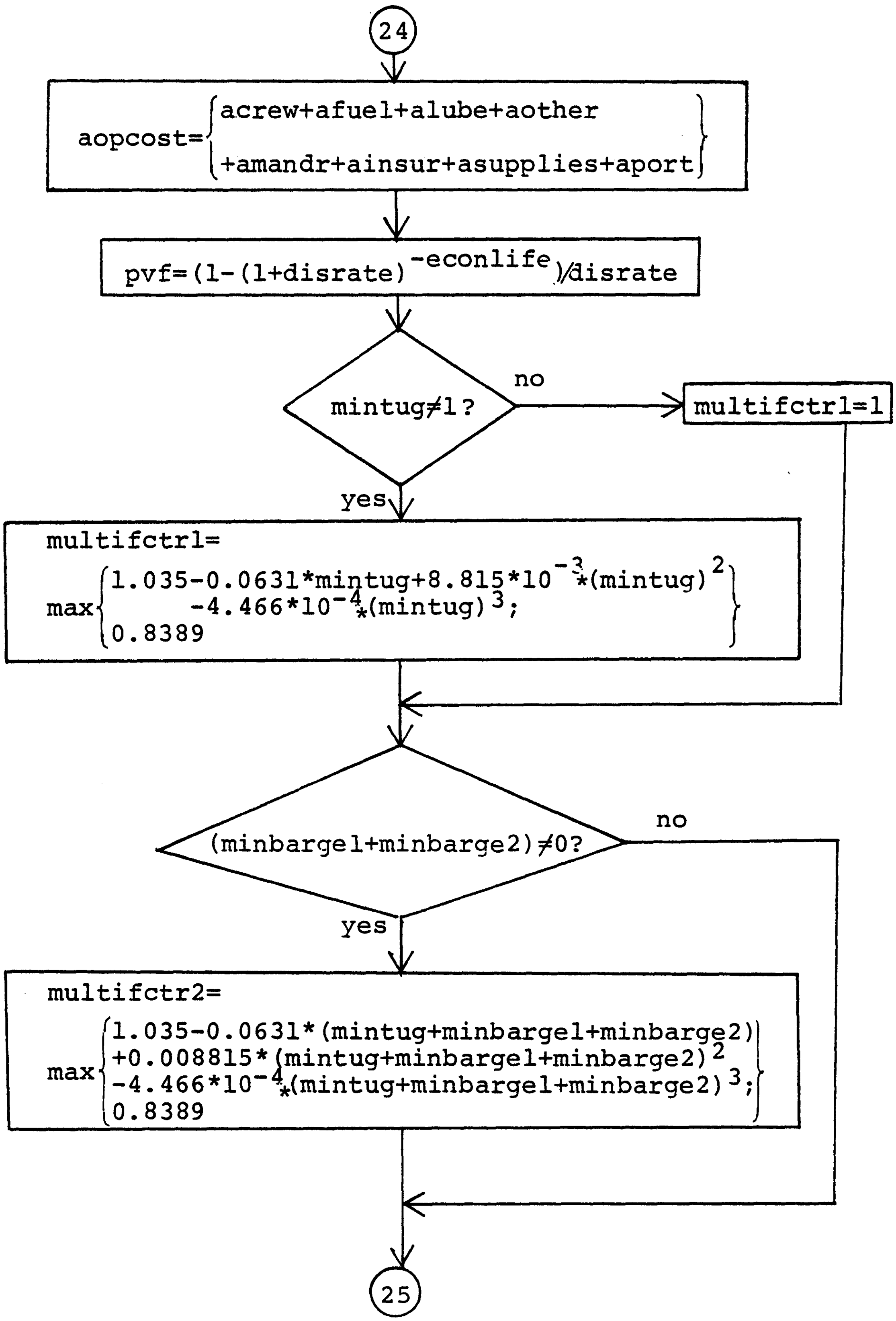
atermop=(aflowave1+aflowave2)*cvarterm

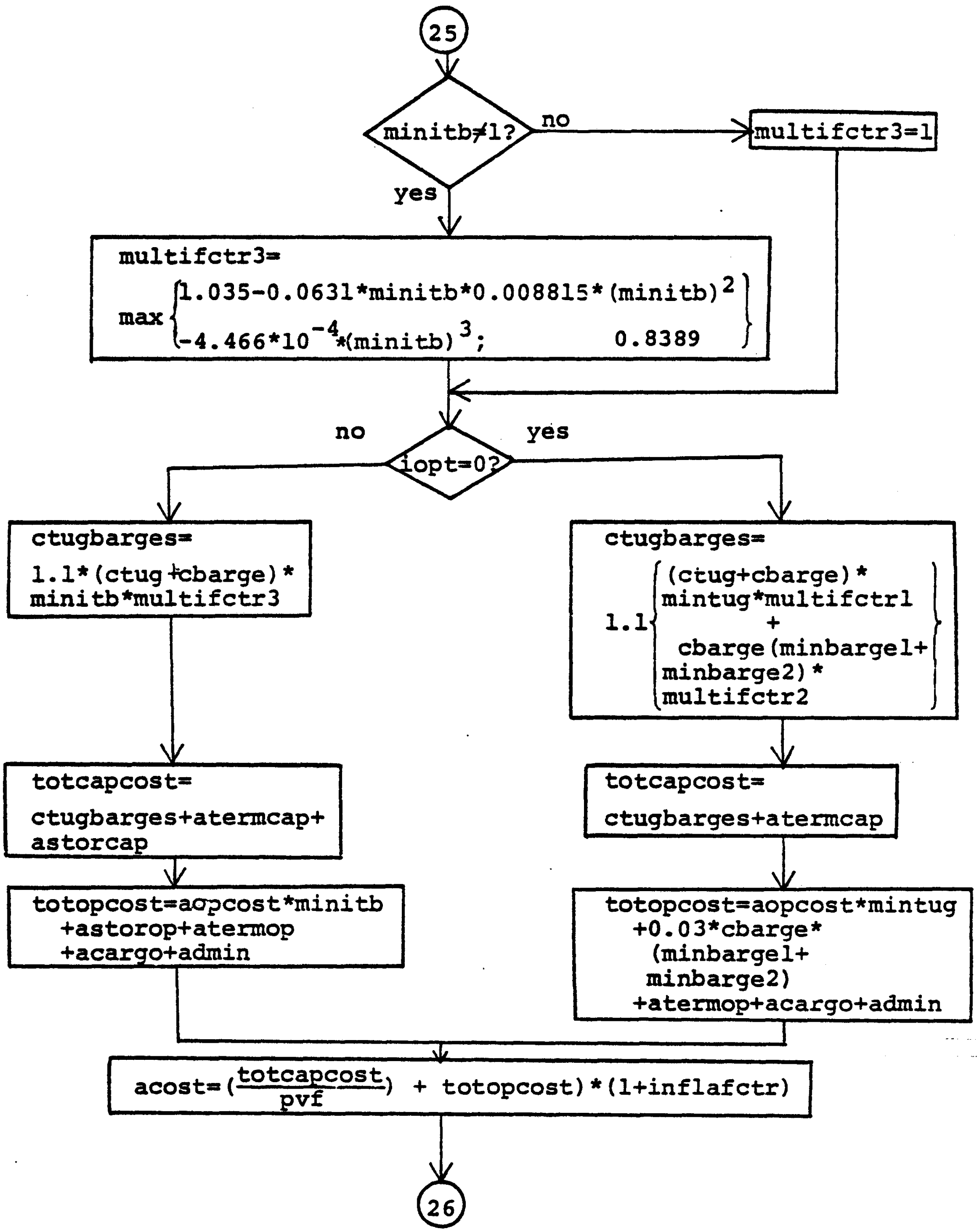
atermcap=
((rload1+runload1)*minbrg1+(rload2+runload2)*minbrg2)
*cfixterm

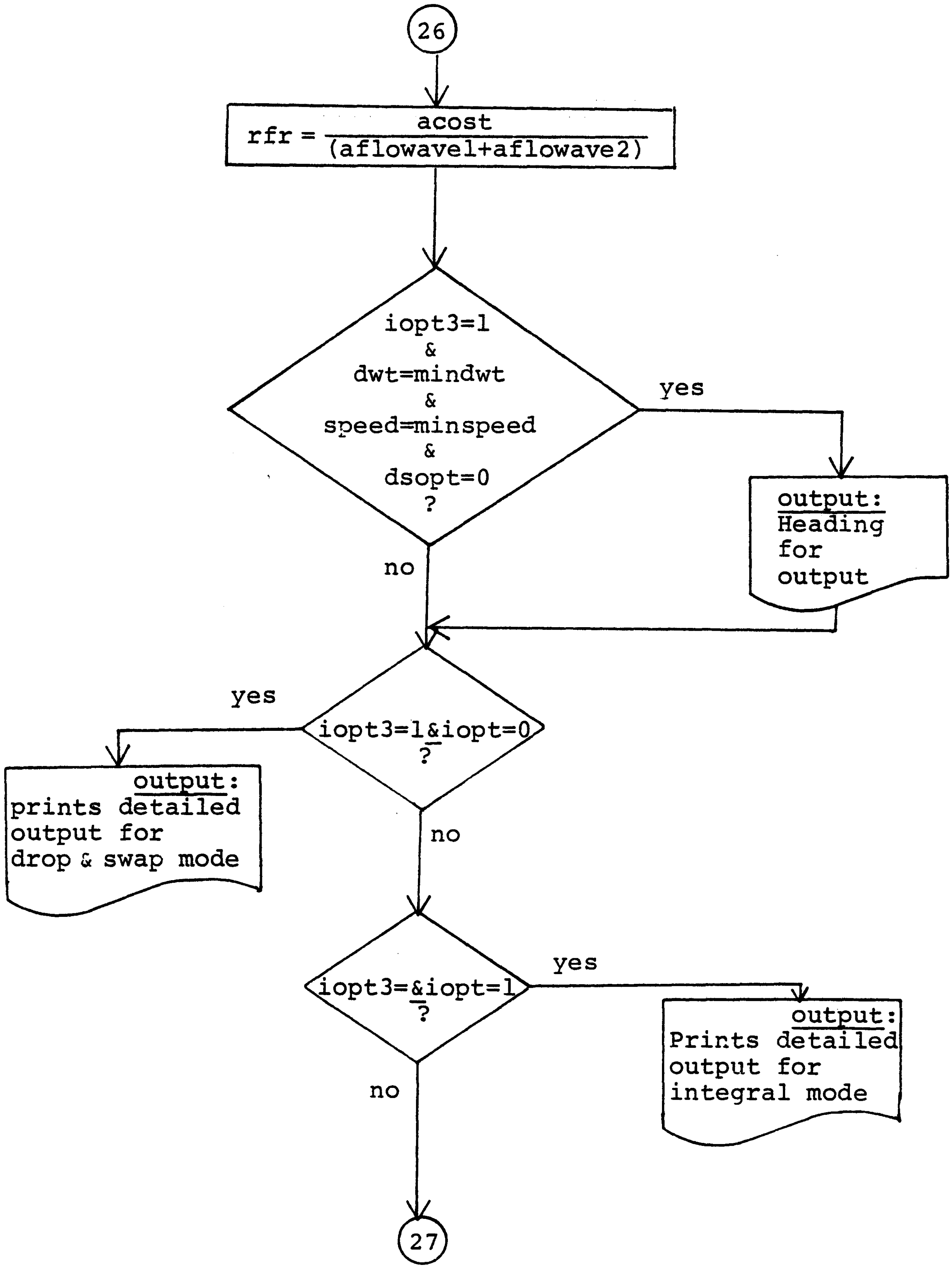
astorop=(aflowave1+aflowave2)*cvarstor

astorcap=dwt*(minbrg1+minbrg2)*cfixstor

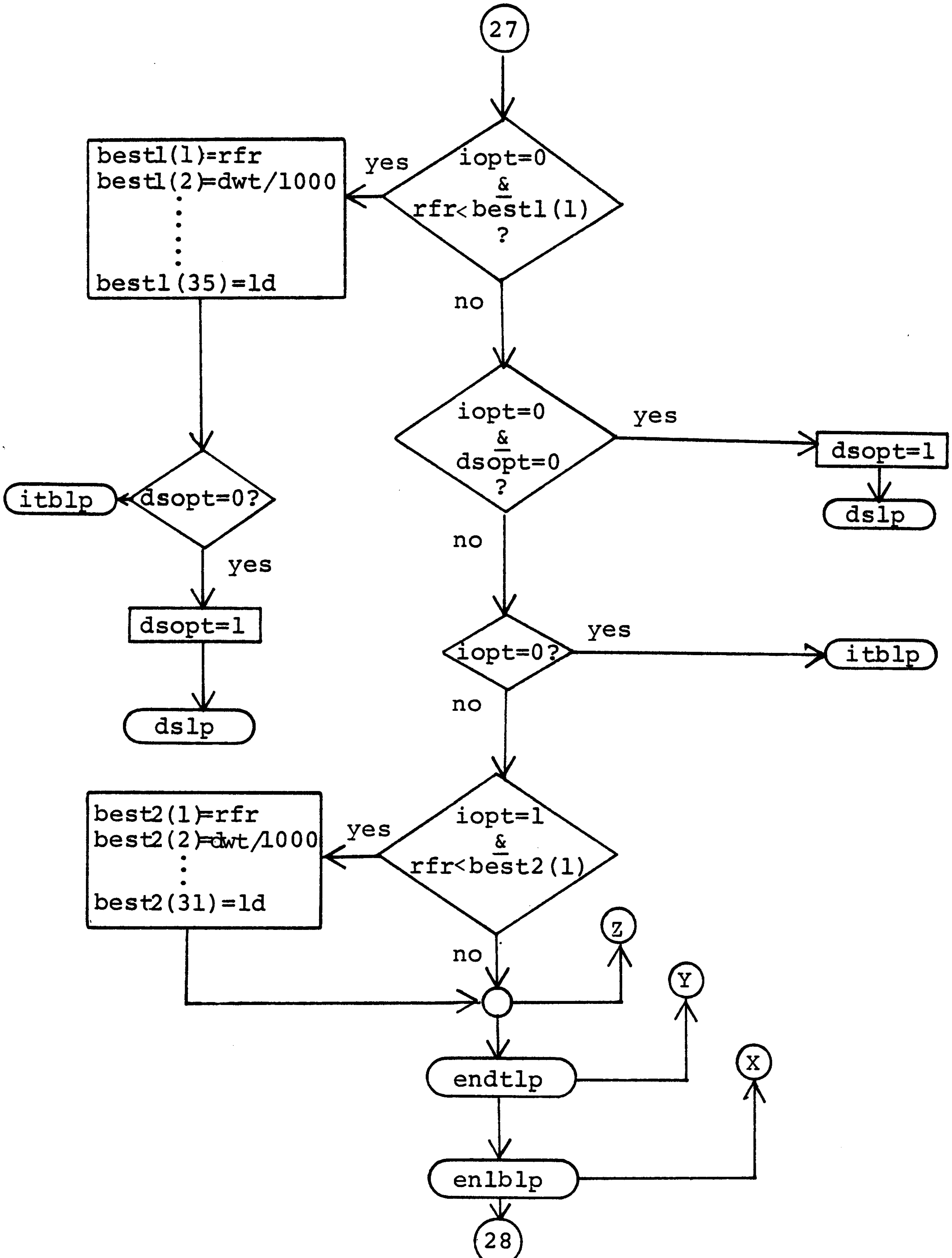
24

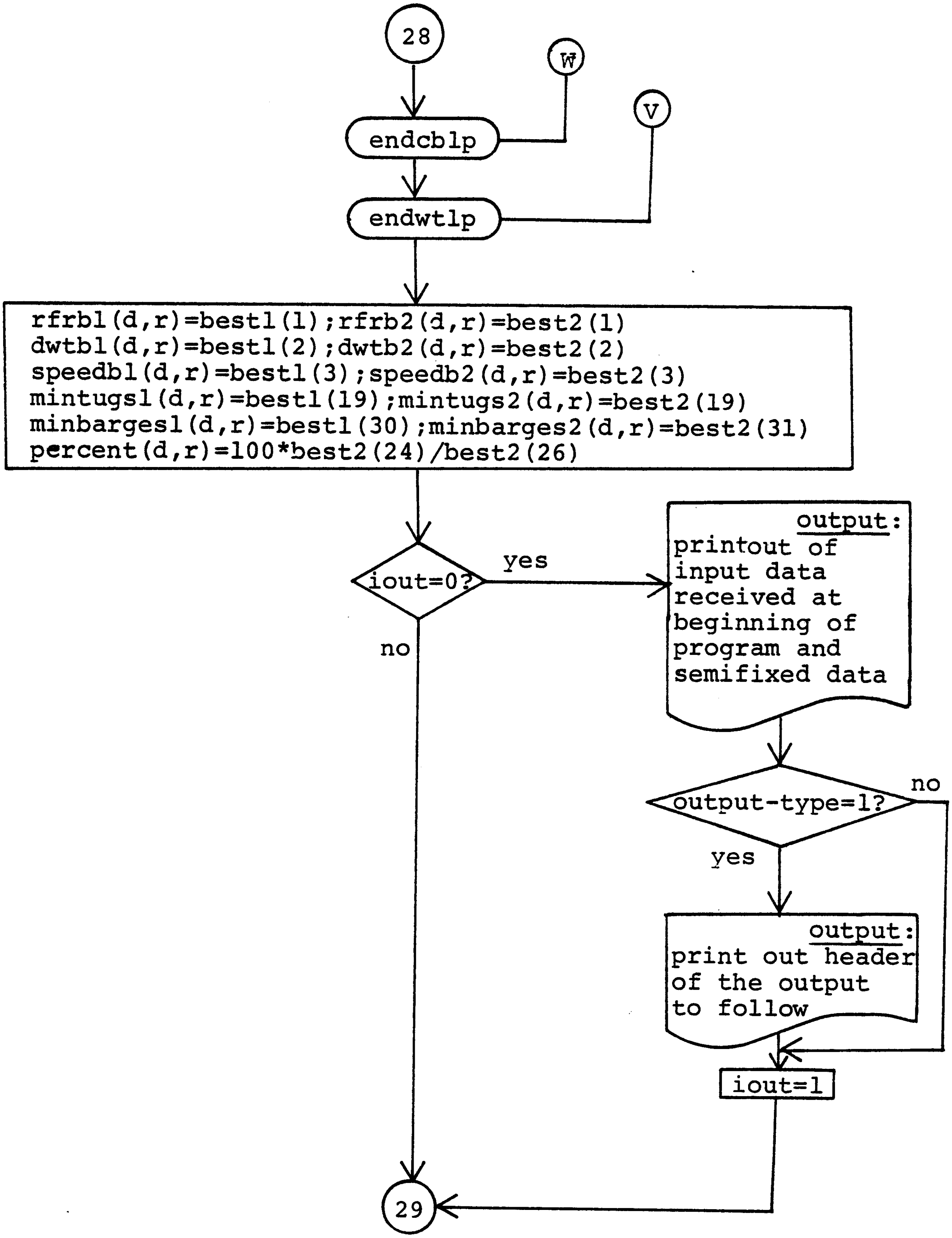

24
aopcost=
acrew+afuel+alube+aother
+amandr+ainsur+asupplies+aport
pvf=(1-(1+disrate)-econlife)/disrate
mintug≠1?
no
multifctrl=1
yes
multifctrl=
max
1.035-0.0631*mintug+8.815*10-3*(mintug)2
-4.466*10-4*(mintug)3;
0.8389
(minbarge1+minbarge2)≠0?
no
yes
multifctr2=
max
1.035-0.0631*(mintug+minbarge1+minbarge2)
+0.008815*(mintug+minbarge1+minbarge2)2
-4.466*10-4*(mintug+minbarge1+minbarge2)3;
0.8389
25

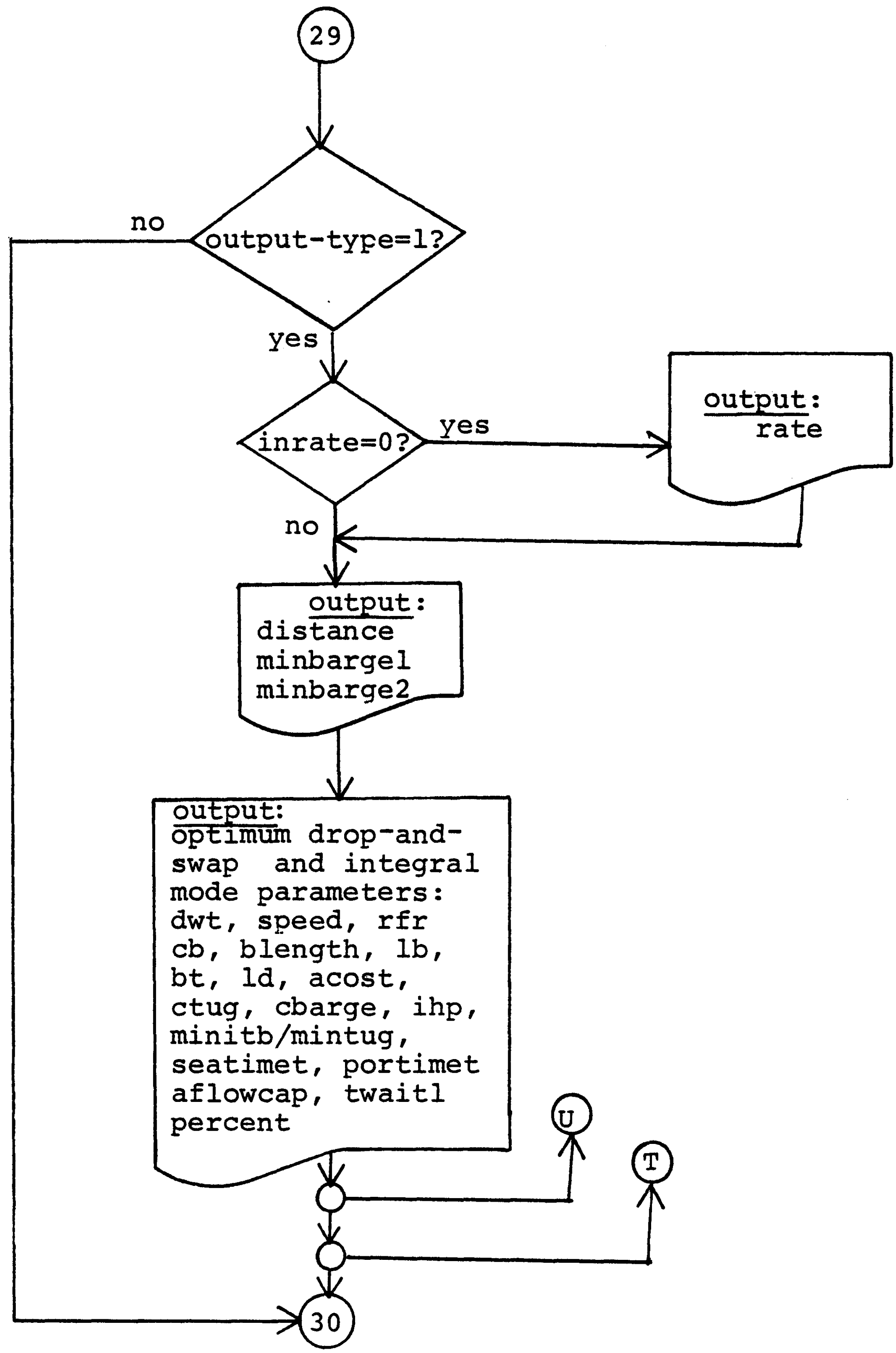

25
minitb≠1?
no
multifctr3=1
yes
multifctr3=
max {1.035-0.0631*minitb*0.008815*(minitb)^2 -4.466*10^-4*(minitb)^3; 0.8389}
no
yes
iopt=0?
ctugbarges=
1.1*(ctug+cbarge)*
minitb*multifctr3
ctugbarges=
1.1{(ctug+cbarge)* mintug*multifctr1 + cbarge(minbarge1+ minbarge2)* multifctr2}
totcapcost=
ctugbarges+atermcap+
astorcap
totcapcost=
ctugbarges+atermcap
totopcost=aopcost*minitb
+astorop+atermop
+acargo+admin
totopcost=aopcost*mintug
+0.03*cbarge*
(minbarge1+
minbarge2)
+atermop+acargo+admin
acost=(totcapcost/pvf) + totopcost)*(1+inflafctr)
26

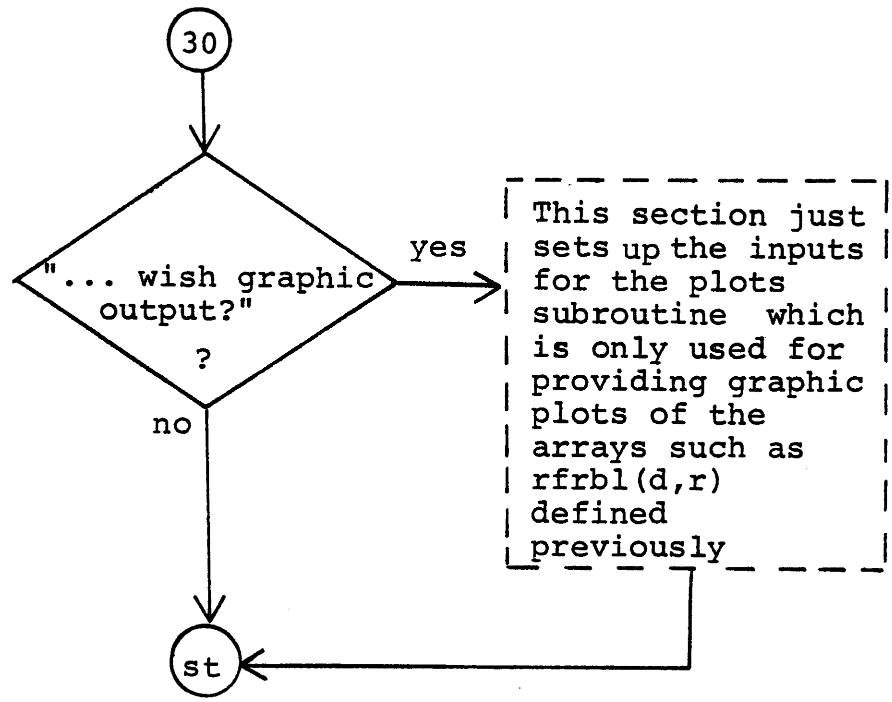


26
rfr = acost / (aflowave1+aflowave2)
iopt3=1
&
dwt=mindwt
&
speed=minspeed
&
dsopt=0
?
yes
output:
Heading
for
output
no
iopt3=1&iopt=0
?
yes
output:
prints detailed
output for
drop & swap mode
no
iopt3=&iopt=1
?
yes
output:
Prints detailed
output for
integral mode
no
27


27
iopt=0
&
rfr<bestl(1)
?
yes
bestl(1)=rfr
bestl(2)=dwt/1000
bestl(35)=ld
no
dsopt=0?
itblp
yes
dsopt=1
dslp
iopt=0
&
dsopt=0
?
yes
dsopt=1
dslp
no
iopt=0?
yes
itblp
no
iopt=1
&
rfr<best2(1)
yes
best2(1)=rfr
best2(2)=dwt/1000
best2(31)=ld
no
Z
Y
X
endtlp
enlblp
28


28
W
V
endcblp
endwtlp
rfrbl(d,r)=bestl(l);rfrb2(d,r)=best2(l)
dwtbl(d,r)=bestl(2);dwtb2(d,r)=best2(2)
speedbl(d,r)=bestl(3);speedb2(d,r)=best2(3)
mintugsl(d,r)=bestl(19);mintugs2(d,r)=best2(19)
minbargesl(d,r)=bestl(30);minbarges2(d,r)=best2(31)
percent(d,r)=100*best2(24)/best2(26)
iout=0?
yes
no
output:
printout of input data received at beginning of program and semifixed data
output-type=1?
no
yes
output:
print out header of the output to follow
iout=1
29


29
output-type=1?
no
yes
inrate=0?
yes
output: rate
no
output: distance minbarge1 minbarge2
output: optimum drop-and-swap and integral mode parameters: dwt, speed, rfr cb, blength, lb, bt, ld, acost, ctug, cbarge, ihp, minitb/mintug, seatimet, portimet aflowcap, twaitl percent
U
T
30


30
"... wish graphic output?" ?
yes
no
This section just sets up the inputs for the plots subroutine which is only used for providing graphic plots of the arrays such as rfrbl(d,r) defined previously
st

```
drop_and_swap: proc;
dcl(rfrb1, rfrb2, dwtb1, dwtb2, speedb1, speedb2, mintugs1, mintugs2,
     minbarges1, minbarges2, percent)(25, 50) float bin;
dcl(distance, speed, dwt, tdwt, maxl, maxb, maxt1, maxt2, rate,
     ltug, lbarge, litb, bbarge, dbarge, tbarge, displ,
     rload1, rload2, runload1, runload2, delay1, delay2,
     aflowave1, aflowave2, mflowave1, mflowave2,
     tugopdays, bargeopdays, tlink, tunlink, twait1, twait2,
     servmargin, fuelmargin, sfc, temp, shp, ihp,
     resist, fuelcons, lubecons, multifctr1, multifctr2, multifctr3,
     wsteelt, woutfitt, wmacht, wmisc, csteelt, coutfitt, cmacht,
     nrcrew, cwages, csubs, ctug, cbarge, ctugbarges, cfuel, clube,
     cfixport1, cfixport2, cvarport1, cvarport2,
     cvarstor, cfixstor, cvarterm, cfixterm,
     minbarge1, minbarge2, minbrg1, minbrg2, mintug, minitb, tport1, tport2,
     rseafuel, rlubeoil, rportfuel, seatimet, portimet, ttript,
     disrate, pvf, inflafctr, vcargo, econlife, rfr, nrtrips, best1(50), best2(50),
     acrew, afuel, alube, amandr, ainsur, asupplies, aport, atermop, atermcap, astorop, astorcap, acargo, aflowcap,
     admin, aother, aopcost, totopcost, totcapcost, acost) float bin;
dcl(llcoef(66), c1(39), c2(47), dt(9), dq(13), rc(2, 8, 7, 8), t(15, 15), w(15, 15), eta(15, 15)) float bin;
dcl(lbarge75, lbarge80, lbarge85, xcb, x1cb, x2cb, x3cb, dhp, wa, th, hr, propef, ehp) float bin;
dcl(hullwt75, hullwt80, hullwt85, wsteelb, woutfitb, csteelb, coutfitb) bin float;
dcl(mincb, maxcb, delcb, minlb, maxlb, dellb, minbt, maxbt, delbt, lb, bt, cb) dec float;
dcl(minfbd, ldfact, shrfact, fbd, ld, volwt) bin float;
dcl(minspeed, maxspeed, delspeed, mindwt, maxdwt, deldwt, mindwttemp, mindist, maxdist, deldist,
     minrate, maxrate, delrate, drange, rrange, d, r, countr) fixed bin;
dcl(iopt, i, j, k, m, iout, iout2, iout3, iopt3, output_type, inrate, dsopt, index) fixed bin;
dcl  nextchar char(1);
dcl  power entry(bin float, bin float, bin float, bin float, bin float, bin float, bin float,
     bin float, bin float, bin float, bin float, bin fixed, dim(39) bin float,
     dim(47) bin float, dim(9) bin float, dim(13) bin float, dim(2, 8, 7, 8) bin float,
     dim(15, 15) bin float, dim(15, 15) bin float, dim(15, 15) bin float) external;
dcl  ibit fixed bin static;
dcl(parameters, llfact, powerdata, sysin) file stream input, sysprint file stream output;
open file(llfact) title("vfile_ loadline.data");
get file(llfact) list(llcoef);
close file(llfact);
open file(powerdata) title("vfile_ power.data");
get skip(4) file(powerdata) edit((((rc(i, j, k, m) do m = 1 to 8) do k = 1 to 7) do j = 1 to 8) do i = 1 to 2))
  (skip(2), 7(col(1), x(5), 8 f(5)));
get skip(5) file(powerdata) edit(((w(i, j) do j = 1 to 15) do i = 1 to 15))(col(1), x(5), 15 f(5));
get skip(5) file(powerdata) edit(((eta(i, j) do j = 1 to 15) do i = 1 to 15))(col(1), x(5), 15 f(6));
get skip(5) file(powerdata) edit(((t(i, j) do j = 1 to 15) do i = 1 to 15))(col(1), x(5), 15 f(5));
get skip(3) file(powerdata) list(c1, c2, dt, dq);
close file(powerdata);
/* --------------- BEGIN include file plot_entry_dcls.incl.pl1 --------------- */

dcl  plot_          ext entry (float bin dimension (*), float bin dimension (*), fixed bin, fixed bin, char (1)),
     plot_$setup    ext entry (char (*), char (*), char (*), fixed bin, float bin, fixed bin, fixed bin),
     plot_$scale    ext entry (float bin, float bin, float bin, float bin);

dcl (Vectors_only initial (1),                    /* vec_sw;  fourth arg to plot_ */
     Vectors_and_symbols initial (2),
     Symbols_only initial (3),

     Linear_linear initial (1),                   /* type;  fourth arg to plot_$setup */
     Log_linear initial (2),
     Linear_log initial (3),
     Log_log initial (4),

```

```
     Tick_marks initial (0),                  /* grid_sw;  fifth arg to plot_$setup */
     Dotted_grid initial (1),
     Solid_grid initial (2),
     No_grid initial (3),

     Normal_scaling initial (0),              /* eq_scale_sw;  sixth arg to plot_$setup */
     Equal_scaling initial (1)) fixed bin static options (constant);

/* ---------------- END include file plot_entry_dcls.incl.pl1 ---------------- */

          /* input basic changeable parameters via get list statements */
st: put list("Input via list format the following parameters:");
put skip(2) edit("Do you wish to specify individual port L-D rates?: ")
  (col(1), a);
call yesno;
inrate = ibit;
if inrate = 1 then do;
   put skip edit("rload1, runload1, rload2, runload2:  ")(a);
   get list(rload1, runload1, rload2, runload2);
   delrate = 1;
   minrate = 0; maxrate = 0;
   go to sp;
end;
put skip edit("minrate, maxrate, delrate:  ")(a);
get list(minrate, maxrate, delrate);
if delrate = 0 then delrate = 1000;
sp: put skip edit("mindist, maxdist, deldist:  ")(a);
get list(mindist, maxdist, deldist);
if deldist = 0 then deldist = 1000;
put skip edit("minspeed, maxspeed, delspeed:  ")(a);
get list(minspeed, maxspeed, delspeed);
if delspeed = 0 then delspeed = 1;
put skip edit("mindwt, maxdwt, deldwt:  ")(a);
get list(mindwt, maxdwt, deldwt);
if deldwt = 0 then deldwt = 5000;
mindwttemp = mindwt;
put skip edit("aflowave1, aflowave2:  ")(a);
get list(aflowave1, aflowave2);
if aflowave1 = 0.e0 then do;
   rload1 = 0.e0;
   runload2 = 0.e0;
end;
if aflowave2 = 0.e0 then do;
   rload2 = 0.e0;
   runload1 = 0.e0;
end;
          /* input form parameters of tug-barge system */
st2:
put skip edit("mincb, maxcb, delcb:  ")(a);
get list(mincb, maxcb, delcb);
if delcb = 0.e0 then delcb = 0.1;
put skip edit("minlb, maxlb, dellb:     ")(a);
get list(minlb, maxlb, dellb);
if dellb = 0.e0 then dellb = 0.2;
put skip edit("minbt, maxbt, delbt:    ")(a);
get list(minbt, maxbt, delbt);
if delbt = 0.e0 then delbt = 0.1;
if minbt<2.0 | minbt>3.25 | maxbt>3.25 | maxbt<2.0 | mincb<0.75 | mincb>0.85 |
maxcb<0.75 | maxcb>0.85 | minlb>8.0 | maxlb>8.0 then do;
   put skip list("Form coefficients are out of interpolable range.  Please specify:");
```

```
   put skip list("0.75-cb-0.85    6.0-L/B-8.0     2.0-B/T-3.25");
   iout3 = 1;
   go to st2;
end;
if iout3 = 1 then do;
   iout3 = 0;
   go to st3;
end;
         /* input basic semifixed parameter values via parameter file */
open file(parameters) title("vfile_ semifixedparams.data");
get file(parameters) edit(tugopdays, bargeopdays, tlink, tunlink,
   servmargin, fuelmargin, sfc, cfuel, clube, nrcrew,
   cwages, csubs, csteelt, coutfitt, csteelb, coutfitb, ltug, wmisc, aother,
   admin, cfixport1, cfixport2, cvarport1, cvarport2, cfixterm, cvarterm, cfixstor, cvarstor,
   delay1, delay2, maxl, maxb, maxt1, maxt2,
   disrate, econlife, inflafctr, vcargo)(skip(3), 38(col(21), f(12, 2)));
close file(parameters);
put skip edit("input changes to semifixed data via get data format")(a);
get data(tugopdays, bargeopdays, tlink, tunlink,
   servmargin, fuelmargin, sfc, cfuel, clube, nrcrew,
   cwages, csubs, csteelt, coutfitt, csteelb, coutfitb, ltug, wmisc, aother,
   admin, cfixport1, cfixport2, cvarport1, cvarport2, cfixterm, cvarterm, cfixstor, cvarstor,
   delay1, delay2, maxl, maxb, maxt1, maxt2,
   disrate, econlife, inflafctr, vcargo);
put skip edit("Do you want printed output?: ")(col(1), a);
call yesno;
output_type = ibit;
if output_type = 1 then do;
   put skip edit("Do you want detailed output?: ")(col(1), a);
   call yesno;
   iopt3 = ibit;
end;
         /* this segment initializes various variables */
st3: iout = 0;
if(maxdist - mindist) < 1 | deldist < 1 then drange = 1;
else drange =(maxdist - mindist)/deldist + 1;
if(maxrate - minrate) < 1 | delrate < 1 then rrange = 1;
else rrange =(maxrate - minrate)/delrate + 1;
do distance = mindist to maxdist by deldist;
   d =(distance - mindist)/deldist + 1;
   do rate = minrate to maxrate by delrate;
      r =(rate - minrate)/delrate + 1;
      if inrate = 0 then do;
         rload1 = rate;
         rload2 = rate;
         runload1 = rate;
         runload2 = rate;
      end;
      iout2 = 0;
      if rate ^= minrate then
         mindwt = max(1.0e3*min(best1(2), best2(2))-2.*deldwt, mindwttemp);
      else mindwt = mindwttemp;
      best1 = 0.0e0;
      best2 = 0.0e0;
      best1(1) = 9.99e2;
      best2(1) = 9.99e2;
      mflowave1 =(aflowave1*3.65e2)/(1.2e1*bargeopdays);
      mflowave2 =(aflowave2*3.65e2)/(1.2e1*bargeopdays);
      if rload1 = 0.e0 | runload1 = 0.e0 then
         if rload1 = 0.e0 then minbarge1 = ceil(mflowave2/(3.05e1*runload1));
```

```
         else minbarge1 = ceil(mflowave1/(3.05e1*rload1));
      else minbarge1 = ceil(mflowave1/(3.05e1*rload1) + mflowave2/(3.05e1*runload1));
      if rload2 = 0.e0 | runload2 = 0.e0 then
         if rload2 = 0.e0 then minbarge2 = ceil(mflowave1/(3.05e1*runload2));
         else minbarge2 = ceil(mflowave2/(3.05e1*rload2));
      else minbarge2 = ceil(mflowave2/(3.05e1*rload2) + mflowave1/(3.05e1*runload2));
      minbrg1 = minbarge1; minbrg2 = minbarge2;
         /* outermost loop-iterate barge size */
dwtlp:
      do dwt = mindwt to maxdwt by deldwt;
         /* in this section the cost of the barge and its principal dimensions must be */
         /* determined as a function of cdwt.   For the time being empirical relations */
         /* will be used until a formal costing routine is developed.                  */
         do cb = mincb to maxcb by delcb;
            xcb = 2.*(cb - 0.75);
            x1cb = 1.0 +(xcb-3.0)*xcb/2.0;
            x2cb =(xcb-2.0)*xcb;
            x3cb =(xcb-1.0)*xcb/2.0;
            do lb = minlb to maxlb by dellb;
               do bt = minbt to maxbt by delbt;
                  lbarge75 = exp(1.128)*lb**0.742*bt**0.382*dwt**0.336;
                  lbarge80 = exp(1.108)*lb**0.739*bt**0.381*dwt**0.336;
                  lbarge85 = exp(1.088)*lb**0.737*bt**0.379*dwt**0.336;
                  lbarge = lbarge75*x1cb - lbarge80*x2cb + lbarge85*x3cb;
                  hullwt75 = exp(-6.206)*bt**0.884*lb**1.348*dwt**1.104;
                  hullwt80 = exp(-6.396)*bt**0.909*lb**1.368*dwt**1.111;
                  hullwt85 = exp(-6.569)*bt**0.930*lb**1.391*dwt**1.117;
                  wsteelb = hullwt75*x1cb - hullwt80*x2cb + hullwt85*x3cb;
                  do j = 1 to 50;
                     bbarge =(lbarge + 0.7*ltug)/lb;
                     tbarge = bbarge/bt;
                     temp = cb*lbarge*bbarge*tbarge/35.;
                     woutfitb = max(1.496*lbarge - 284.24, 50.);
                     displ = woutfitb + wsteelb + dwt;
                     lbarge =(35*displ*(lbarge/bbarge)**2*bt/cb)**0.333333333;
                     if abs(temp - displ) < 1.e0 then go to lloop;
                  end;
                  put skip list("Barge length routine does not converge.");
lloop:
                  if lbarge > 750. then do;
                     put skip edit("barge length exceeds 750'", "lbarge = ", lbarge, "dwt = ", dwt)
                       (a, a, f(6), a, f(6));
                     go to endcblp;
                  end;
                  cbarge = 1.0e3*(wsteelb*csteelb + woutfitb*coutfitb);
                  if lbarge = 750. then minfbd = llcoef(66);
                  else do;
                     countr = ceil((lbarge-99.99999)/10.);
                     minfbd = llcoef(countr) +(llcoef(countr+1)-llcoef(countr))*(lbarge-90.-10.*countr)/10.;
                  end;
                  minfbd = minfbd*(cb + 0.68)/1.38;
                  if(lbarge/(minfbd/12. + tbarge)) < 15. then do;
                     if lbarge<393.6 then ldfact =(tbarge + minfbd/12. - lbarge/15.)*(lbarge/131.2)/(1-lbarge/(12.*131.2))
\c;
                     else ldfact =(tbarge + minfbd/12. - lbarge/15.)*4.;
                  end;
                  else ldfact = 0.;
                  shrfact = 0.0375*lbarge + 3.75;
                  fbd = 0.75*(minfbd + ldfact + shrfact);
                  dbarge =(tbarge + fbd/12.);
                  volwt =(35.*displ + lbarge*bbarge*(dbarge-tbarge) - 2240.*(wsteelb+woutfitb)/490.)/42.;
```

```
                ld = lbarge/dbarge;
                litb = lbarge + 7.e-1*ltug;
                if(tbarge>maxt1) |(tbarge>maxt2) |(litb>maxl) |(bbarge>maxb) |(ld > 16.)
                then go to endbtlp;
          /* intermediate loop-iterate over tug/barge speed */
          /* calcuation of tug IHP */
                if litb/bbarge < 5.99 | litb/bbarge > 8.01 |((litb/bbarge < 6.19 | litb/bbarge > 7.61) &(cb < 0.775 | cb
\c > 0.835))
                then do;
          /*       put skip edit("L/B and/or CB are out of interpolable ranges:  ","L/B = ",lb,",  CB = ",cb,        *
\c/
          /*                       ", litb/bbarge = ",litb/bbarge)(a, 3(a,f(5,2)));          */
                   go to endlblp;
                end;
                do speed = minspeed to maxspeed by delspeed;
          /* this is where the external subroutine power is called to determine dhp    */
                   call power(litb, bbarge, tbarge, cb, speed, wa, th, hr, propef, ehp, dhp, index, c1, c2, dt, dq, rc,
\ct, w, eta);
                   if index = 1 then do;
                      go to endbtlp;
                   end;
                   resist = 550.*ehp/(1.68781*speed);
/* here I am assuming 5% appendage drage and 10% linkage drag    */
                   shp = 1.15*dhp/0.98;
                   ihp =(1.0e0 + servmargin)*shp;
          /* calculate tug cost using univ of michigan study-July 77 --single screw tug */
                   wsteelt =(6.4e-1*(ihp/1.0e3)**2 + 1.679e1*ihp/1.0e3 + 3.78e2);
                   woutfitt =(1.866e-1*(ihp/1.0e3)**2 + 2.733e0*ihp/1.0e3 + 1.54e2);
                   wmacht =(-8.889e-2*(ihp/1.0e3)**2 + 3.288e1*ihp/1.0e3 + 4.999e0);
                   cmacht = 1.3*(3.14e-1*ihp + 1.730e3);
                   ctug =(wsteelt*csteelt + woutfitt*coutfitt + cmacht)*1.0e3;
          /* calculation of annual operating costs */
          /* drop and swap opcosts */
                   iopt = 0;
                   dsopt = 0;
                   if rload1 = 0.e0 | runload1 = 0.e0 then
                      if rload1 = 0.e0 then tport1 = 2.4e1*dwt/runload1;
                      else tport1 = 2.4e1*dwt/rload1;
                   else tport1 =(dwt/rload1 + mflowave2*dwt/(mflowave1*runload1))*2.4e1;
                   if rload2 = 0.e0 | runload2 = 0.e0 then
                      if rload2 = 0.e0 then tport2 = 2.4e1*dwt/runload2;
                      else tport2 = 2.4e1*dwt/rload2;
                   else tport2 =(dwt/runload2 + mflowave2*dwt/(mflowave1*rload2))*2.4e1;
                   seatimet = 2*(distance/speed + tlink + tunlink) + delay1 + delay2;
dslp:              ttript = seatimet/2.4e1;
                   mintug = ceil(max(mflowave1, mflowave2)*ttript/(3.05e1*dwt));
          /* iterative calculation of ttript, twait1, twait2, and mintug */
                   twait1 = 0.0; twait2 = 0.0;
                   do j = 1 to 50;
                      temp = ttript;
                      if dsopt = 0 then do;
                         minbarge1 = minbrg1; minbarge2 = minbrg2;
                         twait1 = max(0.e0, 5.e-1*(mintug*max(tport1/minbarge1, tport2/minbarge2) - seatimet));
                         twait2 = twait1;
                      end;
                      else do;
          /* Calculations for D & S mode with unbalanced trade -- no barges at one port      */
                         if rload1 + runload1 > rload2 + runload2
                         then do;
                            minbarge1 = 0.e0;
                            twait1 = tport1;
                            twait2 = max(0.e0, mintug*(tport2/minbarge2) -(seatimet + tport1));
```

```
                        end;
                        else do;
                           minbarge2 = 0.e0;
                           twait2 = tport2;
                           twait1 = max(0.e0, mintug*(tport1/minbarge1) -(seatimet + tport2));
                        end;
                     end;
                     ttript =(seatimet + twait1 + twait2)/2.4e1;
                     mintug = ceil(max(mflowave1, mflowave2)*ttript/(3.05e1*dwt));
                     if abs(temp - ttript) < 1.0e-3 then go to next3;
                  end;
                  if iout2 = 0 then do;
                     put list("Rttug does not coverge. Divergence is:",(temp-ttript));
                     put data(dwt, speed, rate, distance);
                     iout2 = 1;
                  end;
                  go to itblp;
next3:
                  if(twait1 + twait2) >=(tport1 + tport2) then do;
                     put skip list("Waiting time exceeds porttime. No advantage in drop and swap mode.");
                     put skip data(dwt, speed, rate, distance);
                     go to itblp;
                  end;
                  portimet = twait1 + twait2;
                  go to next4;
      /* integral tug/barge opcosts */
itblp:
                  iopt = 1;
                  seatimet = 2*distance/speed + delay1 + delay2;
                  portimet = tport1 + tport2;
                  ttript =(seatimet + portimet)/2.4e1;
                  minitb = ceil(max(mflowave1, mflowave2)*ttript/(3.05e1*dwt));
next4:
                  rseafuel = shp*sfc/2.240e3;
                  rlubeoil = shp*2.5e-4;
                  rportfuel = 1.25e-1;
                  acrew =(cwages + csubs)*nrcrew;
                  fuelcons = rseafuel*seatimet + rportfuel*portimet;
                  lubecons = rlubeoil*seatimet;
                  nrtrips = tugopdays/ttript;
                  tdwt = dwt +(1.0e0 + fuelmargin)*fuelcons + wmisc;
                  aflowcap = nrtrips*dwt;
                  afuel = fuelcons*nrtrips*cfuel;
                  alube = lubecons*nrtrips*clube;
                  amandr = 1.5*(1.288e2 + 4.539e0*ihp/1.0e3 - 4.0e-2*(ihp/1.0e3)**2 + 2.477e0*tdwt/1.0e3
                     - 9.107e-3*(tdwt/1.0e3)**2)*1.0e3;
                  ainsur = 1.5*(2.10e2 + 3.6e-3*ihp + 1.8e-3*tdwt)*1.0e3;
                  asupplies = 1.3*(4.0e1 + 1.8e-3*ihp + 2.6e-4*tdwt)*1.0e3;
                  aport =(cfixport1 + cfixport2 +(cvarport1 + cvarport2)*dwt)*nrtrips;
                  acargo = vcargo*(aflowave1 + aflowave2)*disrate*seatimet/(48.*365.);
                  atermop = (aflowave1 + aflowave2)*cvarterm;
                  atermcap = ((rload1 + runload1)*minbrg1 + (rload2 + runload2)*minbrg2)*cfixterm;
                  astorop =  (aflowave1 + aflowave2)*cvarstor;
                  astorcap = dwt*(minbrg1 + minbrg2)*cfixstor;
                  aopcost = acrew + afuel + alube + aother + amandr + ainsur
                     + asupplies + aport;
                  pvf =(1.0e0 -(1.0e0 + disrate)**-econlife)/disrate;
      /* compute total annual costs */
                  if mintug ^= 1 then multifctr1 = max(0.8389, 1.035 - 0.0631*mintug + 0.00865*mintug**2
                     - 0.0004466*mintug**3);
                  else multifctr1 = 1.;
```

```
if (minbarge1 + minbarge2) ^= 0 then multifctr2 = max(0.8389, 1.035 - 0.0631*(mintug + minbarge1 + mi
\cnbarge2)
   . + 0.008815*(mintug + minbarge1 + minbarge2)**2 - 0.0004466*(mintug + minbarge1 + minbarge2)**3);
if minitb ^= 1 then multifctr3 = max(0.8389, 1.035 - 0.0631*minitb + 0.008815*minitb**2
   + 0.0004466*minitb**3);
else multifctr3 = 1.;
if iopt = 0 then do;
   ctugbarges = 1.1*((ctug + cbarge)*mintug*multifctr1 + cbarge*(minbarge1 + minbarge2)*multifctr2);
   totcapcost = ctugbarges + atermcap;
   totopcost = aopcost*mintug + 3.e-2*cbarge*(minbarge1 + minbarge2) +  atermop + acargo + admin;
end;
else do;
   ctugbarges = 1.1*(ctug + cbarge)*minitb*multifctr3;
   totcapcost = ctugbarges + atermcap + astorcap;
   totopcost = aopcost*minitb + astorop + atermop + acargo + admin;
end;
acost = (totcapcost/pvf + totopcost)*(1. + inflafctr);
rfr = acost/(aflowave1 + aflowave2);
if iopt3 = 1 & dwt = mindwt & speed = minspeed & dsopt = 0 then
   put skip(3) edit(" DWT   SPD   RFR   CB  BLENGTH  L/B  B/T  L/D   ACOST   CTUG    CBARGE  IHP ",
   "NO_TUG/ITB NRTRIPS SEATIMET PORTIMET AFLOWCAP TWAIT1/$",
   "(1000)(KTS) ($)", "($M)   ($M)    ($M)", "(DAYS)  (DAYS)           (DAYS)")
  (a, a, col(1), a, col(49), a, col(97), a);
if iopt3 = 1 & iopt = 0 then put skip edit(dwt/1.e3, speed, rfr, cb, lbarge, lb, bt, ld, acost/1.e6,
\cctug/1.e6,
   cbarge/1.e6, ihp, mintug, nrtrips, seatimet/24., portimet/24., aflowcap, twait1/24.)
  (x(1), f(3), x(2), f(4,1), x(1), f(6,2), x(1), f(4,2), x(1), f(5,1), x(3), f(4,2), x(1), f(4,2), x(1
\c), f(4,1),
   3(x(1), f(7,3)), x(1), f(5), x(4), f(2), x(6), f(6,2), x(2), f(6,2), x(2), f(7,2), x(2), f(7), x(2)
\c, f(6,2));
else if iopt3 = 1 & iopt = 1 then put skip edit(dwt/1.e3, speed, rfr, cb, lbarge, lb, bt, ld, acost/1
\c.e6,
   ctug/1.e6, cbarge/1.e6, ihp, minitb, nrtrips, seatimet/24., portimet/24., aflowcap, 4.1666*portime
\ct/ttript)
  (x(1), f(3), x(2), f(4, 1), x(1), f(6, 2), x(1), f(4, 2), x(1), f(5, 1), x(3), f(4, 2), x(1), f(4,
\c2), x(1),
   f(4, 1), 3(x(1), f(7, 3)), x(1), f(5), x(4), f(2), x(6), f(6, 2), x(2), f(6, 2), x(2), f(7, 2), x(
\c2), f(7),
   x(2), f(3));
if(iopt = 0) &(rfr < best1(1)) then do;
   best1(1) = rfr;
   best1(2) = dwt/1.e3;
   best1(3) = speed;
   best1(4) = acost/1.e6;;
   best1(5) = ctug/1.e6;
   best1(6) = cbarge/1.e6;
   best1(7) = aopcost;
   best1(8) = totopcost;
   best1(9) = ctugbarges;
   best1(10) = alube;
   best1(11) = afuel;
   best1(12) = amandr;
   best1(13) = ainsur;
   best1(14) = asupplies;
   best1(15) = aport;
   best1(16) = resist;
   best1(17) = ihp;
   best1(18) = nrtrips;
   best1(19) = mintug;
   best1(20) = lbarge;
```

```
                        best1(21) = bbarge;
                        best1(22) = tbarge;
                        best1(23) = displ;
                        best1(24) = portimet/24.;
                        best1(25) = seatimet/24.;
                        best1(26) = ttript;
                        best1(27) = aflowcap;
                        best1(28) = twait1/24.;
                        best1(29) = twait2/24.;
                        best1(30) = minbarge1;
                        best1(31) = minbarge2;
                        best1(32) = cb;
                        best1(33) = lb;
                        best1(34) = bt;
                        best1(35) = ld;
                        if dsopt = 0 then do;
                           dsopt = 1;
                           go to dslp;
                        end;
                        else go to itblp;
                     end;
                     else if(iopt = 0) & dsopt = 0 then do;
                        dsopt = 1;
                        go to dslp;
                     end;
                     else if iopt = 0 then go to itblp;
                     if(iopt = 1) &(rfr < best2(1)) then do;
                        best2(1) = rfr;
                        best2(2) = dwt/1.e3;
                        best2(3) = speed;
                        best2(4) = acost/1.e6;
                        best2(5) = ctug/1.e6;
                        best2(6) = cbarge/1.e6;
                        best2(7) = aopcost;
                        best2(8) = totopcost;
                        best2(9) = ctugbarges;
                        best2(10) = alube;
                        best2(11) = afuel;
                        best2(12) = amandr;
                        best2(13) = ainsur;
                        best2(14) = asupplies;
                        best2(15) = aport;
                        best2(16) = resist;
                        best2(17) = ihp;
                        best2(18) = nrtrips;
                        best2(19) = minitb;
                        best2(20) = lbarge;
                        best2(21) = bbarge;
                        best2(22) = tbarge;
                        best2(23) = displ;
                        best2(24) = portimet/24.;
                        best2(25) = seatimet/24.;
                        best2(26) = ttript;
                        best2(27) = aflowcap;
                        best2(28) = cb;
                        best2(29) = lb;
                        best2(30) = bt;
                        best2(31) = ld;
                     end;
                  end;
endbtlp:       end;
```

```
endlblp:    end;
endcblp: end;
endwtlp: end;
next5:
        rfrb1(d, r) = best1(1); dwtb1(d, r) = best1(2); speedb1(d, r) = best1(3); mintugs1(d, r) = best1(19);
        rfrb2(d, r) = best2(1); dwtb2(d, r) = best2(2); speedb2(d, r) = best2(3); mintugs2(d, r) = best2(19);
        minbarges1(d, r) = best1(30);      percent(d, r) = 100.*best2(24)/best2(26);
        minbarges2(d, r) = best1(31);
        if iout = 0 then do;
           /* output of input parameters in put data format */
           put skip(2) edit("aflowave1  = ", aflowave1, "minrate =      ", minrate, "mindist =       ",
              mindist, "minspeed =    ", minspeed, "mindwt =     ", mindwt)(r(format1));
           put skip edit("aflowave2  = ", aflowave2, "maxrate =      ", maxrate, "maxdist =       ",
              maxdist, "maxspeed =    ", maxspeed, "maxdwt =     ", maxdwt)(r(format1));
           put skip edit("delrate =        ", delrate, "deldist =       ",
              deldist, "delspeed =    ", delspeed, "deldwt =     ", deldwt)
              (col(25), a, f(6), col(50), a, f(6), col(75), a, f(6, 2), col(100), a, f(8));
format1:
           format(col(1), a, f(8), col(25), a, f(6), col(50), a, f(6), col(75), a, f(6, 2), col(100), a, f(8));
           put skip(2) edit("maxl  =        ", maxl, "tugopdays   = ", tugopdays, "servmargin = ",
              servmargin, "csteelt  = ", csteelt, "csteelb   = ", csteelb)(r(format2));
           put skip edit("maxb  =        ", maxb, "bargeopdays = ", bargeopdays, "fuelmargin = ",
              fuelmargin, "coutfitt = ", coutfitt, "coutfitb  = ", coutfitb)(r(format2));
           put skip edit("maxt1 =        ", maxt1, "tlink       = ", tlink, "sfc        = ", sfc,
              "cwages   = ", cwages, "cfixport1 = ", cfixport1)(r(format2));
           put skip edit("maxt2 =        ", maxt2, "tunlink     = ", tunlink, "cfuel      = ", cfuel,
              "csubs    = ", csubs, "cfixport2 = ", cfixport2)(r(format2));
           put skip edit("ltug  =        ", ltug, "delay1      = ", delay1, "clube      = ", clube,
              "nrcrew   = ", nrcrew, "cvarport1 = ", cvarport1)(r(format2));
format2:
           format(col(1), a, f(8, 2), col(25), a, f(6), col(50), a, f(8, 2), col(75), a, f(8, 2), col(100), a, f(8, 2));
           put skip edit("delay2      = ", delay2, "vcargo =      ", vcargo, "cvarport2 = ", cvarport2)(col(25), a, f(6), col
\c(50), a,
                       f(8, 2), col(100), a, f(8, 2));
           put skip(2) edit("aother =    ", aother, "disrate  =     ", disrate, "mincb = ", mincb, "minlb = ", minlb, "minbt =
\c ", minbt,
              "cvarstor =  ", cvarstor) (r(format3));
           put skip edit("admin  =    ", admin, "econlife =     ", econlife, "maxcb = ", maxcb, "maxlb = ", maxlb, "maxbt = ",
\c maxbt,
              "cfixstor =  ", cfixstor) (r(format3));
format3:
           format (col(1), a, f(10,2), col(25), a, f(6,2), col(50), a, f(6,2), col(67), a, f(6,1), col(84), a, f(6,1),
              col(100), a, f(8,2));
           put skip edit("pvf       =     ", pvf, "delcb = ", delcb, "dellb = ", dellb, "delbt = ", delbt, "cvarterm =  ", cva
\crterm)
              (col(25), a, f(6,2), col(50), a, f(6,2), col(67), a, f(6,1), col(84), a, f(6,1), col(100), a, f(8,2));
           put skip edit("inflafctr=    ", inflafctr, "cfixterm =  ", cfixterm) (col(25), a, f(6,2), col(100), a, f(8,2));
           if output_type = 1 then
              put skip(3) edit(" DWT   SPD   RFR   CB  BLENGTH  L/B  B/T  L/D   ACOST   CTUG   CBARGE  IHP ",
              "NO TUG/ITB NRTRIPS SEATIMET PORTIMFT AFLOWCAP TWAIT1/%",
              "(1000)(KTS) ($)", "($M)   ($M)    ($M)", "(DAYS)  (DAYS)          (DAYS)")
             (a, a, col(1), a, col(49), a, col(97), a);
           iout = 1;
        end;
        if output_type = 1 then do;
           if inrate = 0 then put skip(2) edit("rate = ", rate)(a, f(8));
           put edit("distance = ", distance, "minbarge1 = ", best1(30),
              "minbarge2 = ", best1(31))(col(1), 3(x(2), a, f(8)));
           put skip edit(best1(2), best1(3), best1(1), best1(32), best1(20), best1(33), best1(34), best1(35), best1(4), best
\c1(5),
              best1(6), best1(17), best1(19), best1(18), best1(25), best1(24), best1(27), best1(29))
```

```
              (x(1), f(3), x(2), f(4, 1), x(1), f(6, 2), x(1), f(4, 2), x(1), f(5, 1), x(3), f(4, 2), x(1), f(4, 2), x(1), f(
\c4, 1),
               3(x(1), f(7, 3)), x(1), f(5), x(4), f(2), x(6), f(6, 2), x(2), f(6, 2), x(2), f(7, 2), x(2), f(7), x(2), f(6,
\c2));
            put skip edit(best2(2), best2(3), best2(1), best2(28), best2(20), best2(29), best2(30), best2(31), best2(4), best
\c2(5),
               best2(6), best2(17), best2(19), best2(18), best2(25), best2(24), best2(27), 100*best2(24)/best2(26))
              (x(1), f(3), x(2), f(4, 1), x(1), f(6, 2), x(1), f(4, 2), x(1), f(5, 1), x(3), f(4, 2), x(1), f(4, 2), x(1), f(
\c4, 1),
               3(x(1), f(7, 3)), x(1), f(5), x(4), f(2), x(6), f(6, 2), x(2), f(6, 2), x(2), f(7, 2), x(2), f(7), x(2), f(3))
\c;
      end;
end; end;
put skip edit("Do you wish graphic output?: ")(col(1), a);
call yesno;
if ibit = 0 then go to st;
put skip edit("Do you wish to plot rfr vs. rate while varying distance?: ")
  (col(1), a);
call yesno;
if ibit = 1 then call plots("REQUIRED FREIGHT RATE VS LOAD/DISCH RATE WHILE VARYING DISTANCE",
   "TONS PER DAY", "RFR-$", rfrb1, rfrb2, 0);
put skip edit("Do you wish to plot rfr vs. distance while varying rate?: ")
  (col(1), a);
call yesno;
if ibit = 1 then call plots("REQUIRED FREIGHT RATE VS DISTANCE WHILE VARYING L-D RATE",
   "MILES", "RFR-$", rfrb1, rfrb2, 1);
put skip edit("Do you wish to plot barge DWT vs. L/D rate while varying distance?: ")
  (col(1), a);
call yesno;
if ibit = 1 then call plots("BARGE DWT VS LOAD/DISCHARGE RATE WHILE VARYING DISTANCE",
   "TONS PER DAY", "DWT", dwtb1, dwtb2, 0);
put skip edit("Do you wish to plot barge DWT vs. distance while varying L/D rate?: ")
  (col(1), a);
call yesno;
if ibit = 1 then call plots("BARGE DWT VS DISTANCE WHILE VARYING L/D RATE",
   "MILES", "DWT", dwtb1, dwtb2, 1);
put skip edit("Do you wish to plot tugspeed vs. L/D rate while varying distance?: ")
  (col(1), a);
call yesno;
if ibit = 1 then call plots("TUG SPEED VS LOAD/DISCHARGE RATE WHILE VARYING DISTANCE",
   "TONS PER DAY", "KTS", speedb1, speedb2, 0);
put skip edit("Do you wish to plot tugspeed vs. distance while varying L/D rate?: ")
  (col(1), a);
call yesno;
if ibit = 1 then call plots("TUG SPEED VS DISTANCE WHILE VARYING L/D RATE",
   "MILES", "KTS", speedb1, speedb2, 1);
put skip edit("Do you wish to plot the minimum no. of required tugs vs L/D rate while varying Distance?: ")
  (col(1), a);
call yesno;
if ibit = 1 then call plots("MIN. NO. OF TUGS REQUIRED VS L/D RATE WHILE VARYING DISTANCE",
   "TONS PER DAY", "NO. | TUGS", mintugs1, mintugs2, 0);
put skip edit("Do you wish to plot the minimum no. of required tugs vs distance while varying L/D rate?: ")
  (col(1), a);
call yesno;
if ibit = 1 then call plots("MIN. NO. OF TUGS REQUIRED VS DISTANCE WHILE VARYING L/D RATE",
   "MILES", "NO. | TUGS", mintugs1, mintugs2, 1);
put skip edit("Do you wish to plot no. of barges required at port 1 and port 2 under D&S ",
   "vs L/D rate?: ")(col(1), a, a);
call yesno;
if ibit = 1 then call plots("MIN NO. BARGES REQUIRED AT PORT 1 AND PORT 2 VS L/D RATE ",
   "TONS PER DAY", "NO. | BARGES", minbarges1, minbarges2, 0);
```

```
put skip edit("Do you wish to plot portime % vs. L/D rate while varying distance?:  ") (col(1), a);
call yesno;
if ibit = 1 then call plots("PORTIME % VS LOAD/DISCHARGE RATE WHILE VARYING DISTANCE",
   "TONS PER DAY", "% PORTIME/TRIPTIME", percent, percent, 0);
go to st;
yesno: proc;
   do j = 1 to 8;
      get edit(nextchar)(a(1));
      if nextchar = "Y" | nextchar = "y" | nextchar = "1" then do;
         ibit = 1;
         get skip;
         return;
      end;
      if nextchar = "N" | nextchar = "n" | nextchar = "0" then do;
         ibit = 0;
         get skip;
         return;
      end;
      ibit = 0;
      get skip;
      return;
   end;
end yesno;
plots: proc(blurb1, blurb2, blurb3, o1, o2, graph_type);
dcl(blurb1, blurb2, blurb3) char(*);
dcl(o1(*, *), o2(*, *)) float bin;
dcl(x, y1, y2)(250) float bin;
dcl(xmin, xmax, ymin, ymax) float bin;
dcl  graph_type fixed bin;
stgr:
   call plot_$setup(blurb1, blurb2, blurb3, 1, 0e0, 1, 0);
   put skip edit("Do you wish to specify graph scales?: ")(col(1), a);
   call yesno;
   if ibit = 1 then do;
      xmin = 0.e0; xmax = 0.e0; ymin = 0.e0; ymax = 0.e0;
      put skip edit("Specify xmin, xmax, ymin, ymax via get data format: ")
        (col(1), a);
      get skip data(xmin, xmax, ymin, ymax);
      call plot_$scale(xmin, xmax, ymin, ymax);
   end;
   if graph_type = 0 then do;
      do d = 1 to drange;
         do r = 1 to rrange;
            x(r) = minrate +(r - 1)*delrate;
            y1(r) = o1(d, r);
            y2(r) = o2(d, r);
         end;
         call plot_(x, y1, rrange, 2, "*");
         call plot_(x, y2, rrange, 2, "+");
   end; end;
   if graph_type = 1 then do;
      do r = 1 to rrange;
         do d = 1 to drange;
            x(d) = mindist +(d - 1)*deldist;
            y1(d) = o1(d, r);
            y2(d) = o2(d, r);
         end;
         call plot_(x, y1, drange, 2, "*");
         call plot_(x, y2, drange, 2, "+");
   end; end;
```

```
   put skip edit("Do you wish to revise plot?: ")(col(1), a);
   call yesno;
   if ibit = 1 then go to stgr;
   return;
end plots;
end drop_and_swap;
```

```
\014      SOURCE FILES USED IN THIS COMPILATION.

LINE      NUMBER  DATE MODIFIED    NAME                            PATHNAME
             0    05/06/79  2349.5  drop__and__swap.pl1             >udd>Barges>Kaskin>drop__and__swap.pl1
46           1    01/27/78  1316.0  plot_entry_dcls.incl.pl1        >ldd>include>plot_entry_dcls.incl.pl1
\014      NAMES DECLARED IN THIS COMPILATION.

IDENTIFIER                OFFSET    LOC STORAGE CLASS   DATA TYPE                ATTRIBUTES AND REFERENCES

NAMES DECLARED BY DECLARE STATEMENT.
acargo                         033275 automatic       float bin(27)            dcl 4 set ref 325 345 350
acost                          033304 automatic       float bin(27)            dcl 4 set ref 352 353 359 363 372 419
acrew                          033262 automatic       float bin(27)            dcl 4 set ref 312 330
admin                          033277 automatic       float bin(27)            dcl 4 set ref 107 115 345 350 485
aflowave1                      033014 automatic       float bin(27)            dcl 4 set ref 74 75 153 325 326 328 353 460
aflowave2                      033015 automatic       float bin(27)            dcl 4 set ref 74 79 154 325 326 328 353 462
aflowcap                       033276 automatic       float bin(27)            dcl 4 set ref 317 359 363 395 442
afuel                          033263 automatic       float bin(27)            dcl 4 set ref 318 330 379 426
ainsur                         033266 automatic       float bin(27)            dcl 4 set ref 322 330 381 428
alube                          033264 automatic       float bin(27)            dcl 4 set ref 319 330 378 425
amandr                         033265 automatic       float bin(27)            dcl 4 set ref 320 330 380 427
aopcost                        033301 automatic       float bin(27)            dcl 4 set ref 330 345 350 375 422
aother                         033300 automatic       float bin(27)            dcl 4 set ref 107 115 330 483
aport                          033270 automatic       float bin(27)            dcl 4 set ref 324 330 383 430
astorcap                       033274 automatic       float bin(27)            dcl 4 set ref 329 349
astorop                        033273 automatic       float bin(27)            dcl 4 set ref 328 350
asupplies                      033267 automatic       float bin(27)            dcl 4 set ref 323 330 382 429
atermcap                       033272 automatic       float bin(27)            dcl 4 set ref 327 344 349
atermop                        033271 automatic       float bin(27)            dcl 4 set ref 326 345 350
bargeopdays                    033021 automatic       float bin(27)            dcl 4 set ref 107 115 153 154 471
bbarge                         033002 automatic       float bin(27)            dcl 4 set ref 186 187 188 191 216 219 223 223
                                                                               223 223 231 389 436
best1                          033116 automatic       float bin(27)            array dcl 4 set ref 146 149 151 368 369 370 371
                                                                               372 373 374 375 376 377 378 379 380 381 382 383
                                                                               384 385 386 387 388 389 390 391 392 393 394 395
                                                                               396 397 398 399 400 401 402 403 453 454 454 454
                                                                               456 457 502 502 504 504 504 504 504 504 504 504
                                                                               504 504 504 504 504 504 504 504 504 504
best2                          033200 automatic       float bin(27)            array dcl 4 set ref 146 150 152 415 416 417 418
                                                                               419 420 421 422 423 424 425 426 427 428 429 430
                                                                               431 432 433 434 435 436 437 438 439 440 441 442
                                                                               443 444 445 446 455 455 455 455 456 456 508 508
                                                                               508 508 508 508 508 508 508 508 508 508 508 508
                                                                               508 508 508 508 508
blurb1                                parameter       char                     unaligned dcl 586 set ref 585 591
blurb2                                parameter       char                     unaligned dcl 586 set ref 585 591
blurb3                                parameter       char                     unaligned dcl 586 set ref 585 591
bt                             036710 automatic       float dec(10)            dcl 23 set ref 176 177 178 179 181 182 183 187
                                                                               191 359 363 402 445
c1                             033407 automatic       float bin(27)            array dcl 20 set ref 44 231
c2                             033456 automatic       float bin(27)            array dcl 20 set ref 44 231
cb                             036713 automatic       float dec(10)            dcl 23 set ref 170 171 188 191 207 223 223 231
                                                                               231 359 363 400 443
cbarge                         033055 automatic       float bin(27)            dcl 4 set ref 201 343 343 345 348 359 363 374
                                                                               421
cfixport1                      033061 automatic       float bin(27)            dcl 4 set ref 107 115 324 473
cfixport2                      033062 automatic       float bin(27)            dcl 4 set ref 107 115 324 475
cfixstor                       033066 automatic       float bin(27)            dcl 4 set ref 107 115 329 485
cfixterm                       033070 automatic       float bin(27)            dcl 4 set ref 107 115 327 492
cfuel                          033057 automatic       float bin(27)            dcl 4 set ref 107 115 318 475
```

clube	033060 automatic	float bin(27)	dcl 4 set ref 107 115 319 477
cmacht	033050 automatic	float bin(27)	dcl 4 set ref 243 244
countr	036745 automatic	fixed bin(17,0)	dcl 25 set ref 204 205 205 205 205
coutfitb	036651 automatic	float bin(27)	dcl 22 set ref 107 115 201 471
coutfitt	033047 automatic	float bin(27)	dcl 4 set ref 107 115 244 471
csteelb	036650 automatic	float bin(27)	dcl 22 set ref 107 115 201 469
csteelt	033046 automatic	float bin(27)	dcl 4 set ref 107 115 244 469
csubs	033053 automatic	float bin(27)	dcl 4 set ref 107 115 312 475
ctug	033054 automatic	float bin(27)	dcl 4 set ref 244 343 348 359 363 373 420
ctugbarges	033056 automatic	float bin(27)	dcl 4 set ref 343 344 348 349 377 424
cvarport1	033063 automatic	float bin(27)	dcl 4 set ref 107 115 324 477
cvarport2	033064 automatic	float bin(27)	dcl 4 set ref 107 115 324 481
cvarstor	033065 automatic	float bin(27)	dcl 4 set ref 107 115 328 483
cvarterm	033067 automatic	float bin(27)	dcl 4 set ref 107 115 326 490
cwages	033052 automatic	float bin(27)	dcl 4 set ref 107 115 312 473
d	036743 automatic	fixed bin(17,0)	dcl 25 set ref 136 453 454 454 454 455 455 455 455 456 456 457 603 606 607 614 615 615 616 616 617 617
dbarge	033003 automatic	float bin(27)	dcl 4 set ref 215 216 217
delay1	033012 automatic	float bin(27)	dcl 4 set ref 107 115 257 304 477
delay2	033013 automatic	float bin(27)	dcl 4 set ref 107 115 257 304 481
delbt	036702 automatic	float dec(10)	dcl 23 set ref 92 93 93 176 490
delcb	036660 automatic	float dec(10)	dcl 23 set ref 86 87 87 170 490
deldist	036735 automatic	fixed bin(17,0)	dcl 25 set ref 64 65 65 131 132 135 136 464 615
deldwt	036731 automatic	fixed bin(17,0)	dcl 25 set ref 70 71 71 146 165 464
dellb	036671 automatic	float dec(10)	dcl 23 set ref 89 90 90 175 490
delrate	036740 automatic	fixed bin(17,0)	dcl 25 set ref 56 61 62 62 133 134 137 138 464 605
delspeed	036726 automatic	fixed bin(17,0)	dcl 25 set ref 67 68 68 229 464
dep	036635 automatic	float bin(27)	dcl 21 set ref 231 237
displ	033005 automatic	float bin(27)	dcl 4 set ref 190 191 192 216 391 438
disrate	033107 automatic	float bin(27)	dcl 4 set ref 107 115 325 332 332 483
distance	032766 automatic	float bin(27)	dcl 4 set ref 135 136 257 289 296 304 502
dq	033546 automatic	float bin(27)	array dcl 20 set ref 44 231
drange	036741 automatic	fixed bin(17,0)	dcl 25 set ref 131 132 603 614 619 620
dsopt	036761 automatic	fixed bin(17,0)	dcl 27 set ref 248 264 354 404 405 410 411
dt	033535 automatic	float bin(27)	array dcl 20 set ref 44 231
dwt	032770 automatic	float bin(27)	dcl 4 set ref 165 177 178 179 181 182 183 190 197 249 251 252 252 253 255 256 256 259 284 289 296 307 316 317 324 329 354 359 363 370 417
dwtb1	005004 automatic	float bin(27)	array dcl 2 set ref 454 530 535
dwtb2	007346 automatic	float bin(27)	array dcl 2 set ref 455 530 535
econlife	033113 automatic	float bin(27)	dcl 4 set ref 107 115 332 485
enp	036642 automatic	float bin(27)	dcl 21 set ref 231 235
eta	036265 automatic	float bin(27)	array dcl 20 set ref 42 231
fcd	036721 automatic	float bin(27)	dcl 24 set ref 214 215
fuelcons	033035 automatic	float bin(27)	dcl 4 set ref 313 316 318
fuelmargin	033027 automatic	float bin(27)	dcl 4 set ref 107 115 316 471
graph_type	parameter	fixed bin(17,0)	dcl 590 ref 585 602 612
hr	036640 automatic	float bin(27)	dcl 21 set ref 231
nullwt75	036643 automatic	float bin(27)	dcl 22 set ref 181 184
nullwt80	036644 automatic	float bin(27)	dcl 22 set ref 182 184
nullwt85	036645 automatic	float bin(27)	dcl 22 set ref 183 184
i	036747 automatic	fixed bin(17,0)	dcl 27 set ref 39 39 41 41 42 42 43 43
ibit	000010 internal static	fixed bin(17,0)	dcl 33 set ref 52 123 127 516 520 525 530 535 540 545 550 555 560 564 571 576 580 595 624
inp	033033 automatic	float bin(27)	dcl 4 set ref 238 240 240 241 241 242 242 243 320 320 322 323 359 363 3[illegible] 432
index	036762 automatic	fixed bin(17,0)	dcl 27 set ref 231 232
inflafctr	033111 automatic	float bin(27)	dcl 4 set ref 107 115 352 492

inrate	036760 automatic	fixed bin(17,0)	dcl 27 set ref 52 53 139 501
iopt	036746 automatic	fixed bin(17,0)	dcl 27 set ref 247 302 342 359 363 368 410 414 415
iopt3	036756 automatic	fixed bin(17,0)	dcl 27 set ref 127 354 359 363
iout	036753 automatic	fixed bin(17,0)	dcl 27 set ref 130 458 498
iout2	036754 automatic	fixed bin(17,0)	dcl 27 set ref 145 287 290
iout3	036755 automatic	fixed bin(17,0)	dcl 27 set ref 98 101 102
j	036750 automatic	fixed bin(17,0)	dcl 27 set ref 39 39 41 41 42 42 43 43 185 262 568
k	036751 automatic	fixed bin(17,0)	dcl 27 set ref 39 39
lb	036705 automatic	float dec(10)	dcl 23 set ref 175 177 178 179 181 182 183 186 359 363 401 444
lbarge	033000 automatic	float bin(27)	dcl 4 set ref 180 186 188 189 191 191 195 197 202 204 205 208 209 209 209 209 210 213 216 217 218 359 363 388 435
lbarge75	036626 automatic	float bin(27)	dcl 21 set ref 177 180
lbarge80	036627 automatic	float bin(27)	dcl 21 set ref 178 180
lbarge85	036630 automatic	float bin(27)	dcl 21 set ref 179 180
ld	036722 automatic	float bin(27)	dcl 24 set ref 217 219 359 363 403 446
ldfact	036717 automatic	float bin(27)	dcl 24 set ref 209 210 212 214
litb	033001 automatic	float bin(27)	dcl 4 set ref 218 219 223 223 223 223 231
llcoef	033305 automatic	float bin(27)	array dcl 20 set ref 36 202 205 205 205
llfact	000024 constant	file	input stream dcl 34 set ref 34 35 36 37
ltug	032777 automatic	float bin(27)	dcl 4 set ref 107 115 186 218 477
lubecons	033036 automatic	float bin(27)	dcl 4 set ref 314 319
m	036752 automatic	fixed bin(17,0)	dcl 27 set ref 39 39
maxb	032773 automatic	float bin(27)	dcl 4 set ref 107 115 219 471
maxbt	036677 automatic	float dec(10)	dcl 23 set ref 92 94 94 176 485
maxcb	036655 automatic	float dec(10)	dcl 23 set ref 86 94 94 170 485
maxdist	036734 automatic	fixed bin(17,0)	dcl 25 set ref 64 131 132 135 462
maxdwt	036730 automatic	fixed bin(17,0)	dcl 25 set ref 70 165 462
maxi	032772 automatic	float bin(27)	dcl 4 set ref 107 115 219 469
maxlb	036666 automatic	float dec(10)	dcl 23 set ref 89 94 175 485
maxrate	036737 automatic	fixed bin(17,0)	dcl 25 set ref 57 61 133 134 137 462
maxspeed	036725 automatic	fixed bin(17,0)	dcl 25 set ref 67 229 462
maxt1	032774 automatic	float bin(27)	dcl 4 set ref 107 115 219 473
maxt2	032775 automatic	float bin(27)	dcl 4 set ref 107 115 219 475
mflowave1	033016 automatic	float bin(27)	dcl 4 set ref 153 157 158 159 162 252 256 259 284 307
mflowave2	033017 automatic	float bin(27)	dcl 4 set ref 154 155 158 161 162 252 256 259 284 307
minbarge1	033071 automatic	float bin(27)	dcl 4 set ref 155 157 158 163 265 266 273 280 337 337 337 337 343 345 398
minbarge2	033072 automatic	float bin(27)	dcl 4 set ref 159 161 162 163 265 266 275 278 337 337 337 337 343 345 399
minbarges1	023520 automatic	float bin(27)	array dcl 2 set ref 456 560
minbarges2	026062 automatic	float bin(27)	array dcl 2 set ref 457 560
minbrg1	033073 automatic	float bin(27)	dcl 4 set ref 163 265 327 329
minbrg2	033074 automatic	float bin(27)	dcl 4 set ref 163 265 327 329
minbt	036674 automatic	float dec(10)	dcl 23 set ref 92 94 94 176 483
mincb	036652 automatic	float dec(10)	dcl 23 set ref 86 94 94 170 483
mindist	036733 automatic	fixed bin(17,0)	dcl 25 set ref 64 131 132 135 136 460 615
mindwt	036727 automatic	fixed bin(17,0)	dcl 25 set ref 70 72 146 148 165 354 460
mindwttemp	036732 automatic	fixed bin(17,0)	dcl 25 set ref 72 146 148
minfbd	036716 automatic	float bin(27)	dcl 24 set ref 202 205 207 207 208 209 210 214
minitb	033076 automatic	float bin(27)	dcl 4 set ref 307 339 339 339 339 348 350 363 434
minlb	036663 automatic	float dec(10)	dcl 23 set ref 89 94 175 483
minrate	036736 automatic	fixed bin(17,0)	dcl 25 set ref 57 61 133 134 137 138 146 460 605
minspeed	036724 automatic	fixed bin(17,0)	dcl 25 set ref 67 229 354 460

mintug	033075 automatic	float bin(27)	dcl 4 set ref 259 266 275 280 284 334 334 334 334 337 337 337 343 345 359 387
mintugs1	016614 automatic	float bin(27)	array dcl 2 set ref 454 550 555
mintugs2	021156 automatic	float bin(27)	array dcl 2 set ref 455 550 555
multifctr1	033037 automatic	float bin(27)	dcl 4 set ref 334 336 343
multifctr2	033040 automatic	float bin(27)	dcl 4 set ref 337 343
multifctr3	033041 automatic	float bin(27)	dcl 4 set ref 339 341 348
nextchar	036763 automatic	char(1)	unaligned dcl 28 set ref 569 570 570 570 575 575 575
nrcrew	033051 automatic	float bin(27)	dcl 4 set ref 107 115 312 477
nrtrips	033115 automatic	float bin(27)	dcl 4 set ref 315 317 318 319 324 359 363 386 433
o1	parameter	float bin(27)	array dcl 587 ref 585 606 616
o2	parameter	float bin(27)	array dcl 587 ref 585 607 617
output_type	036757 automatic	fixed bin(17,0)	dcl 27 set ref 123 124 493 500
parameters	000022 constant	file	input stream dcl 34 set ref 34 106 107 113
percent	030424 automatic	float bin(27)	array dcl 2 set ref 456 564 564
plot_	000034 constant	entry	external dcl 1-3 ref 609 610 619 620
plot_$scale	000040 constant	entry	external dcl 1-3 ref 600
plot_$setup	000036 constant	entry	external dcl 1-3 ref 591
portimet	033105 automatic	float bin(27)	dcl 4 set ref 299 305 306 313 359 363 363 392 439
power	000020 constant	entry	external dcl 29 ref 231
powerdata	000026 constant	file	input stream dcl 34 set ref 34 38 39 41 42 43 44 45
propef	036641 automatic	float bin(27)	dcl 21 set ref 231
pvf	033110 automatic	float bin(27)	dcl 4 set ref 332 352 490
r	036744 automatic	fixed bin(17,0)	dcl 25 set ref 138 453 454 454 454 455 455 455 455 456 456 457 604 605 605 606 606 607 607 613 616 617
rate	032776 automatic	float bin(27)	dcl 4 set ref 137 138 140 141 142 143 146 289 296 501
rc	033563 automatic	float bin(27)	array dcl 20 set ref 39 231
resist	033034 automatic	float bin(27)	dcl 4 set ref 235 384 431
rfr	033114 automatic	float bin(27)	dcl 4 set ref 353 359 363 368 369 415 416
rfrb1	000100 automatic	float bin(27)	array dcl 2 set ref 453 520 525
rfrb2	002442 automatic	float bin(27)	array dcl 2 set ref 455 520 525
rload1	033006 automatic	float bin(27)	dcl 4 set ref 55 76 140 155 155 157 158 249 249 251 252 271 327
rload2	033007 automatic	float bin(27)	dcl 4 set ref 55 80 141 159 159 161 162 253 253 255 256 271 327
rlubeoil	033102 automatic	float bin(27)	dcl 4 set ref 310 314
rportfuel	033103 automatic	float bin(27)	dcl 4 set ref 311 313
rrange	036742 automatic	fixed bin(17,0)	dcl 25 set ref 133 134 604 609 610 613
rseafuel	033101 automatic	float bin(27)	dcl 4 set ref 308 313
runload1	033010 automatic	float bin(27)	dcl 4 set ref 55 81 142 155 155 158 249 249 252 271 327
runload2	033011 automatic	float bin(27)	dcl 4 set ref 55 77 143 159 159 162 253 253 256 271 327
seatimet	033104 automatic	float bin(27)	dcl 4 set ref 257 258 266 275 280 283 304 306 313 314 325 359 363 393 440
servmargin	033026 automatic	float bin(27)	dcl 4 set ref 107 115 238 469
sfc	033030 automatic	float bin(27)	dcl 4 set ref 107 115 308 473
shp	033032 automatic	float bin(27)	dcl 4 set ref 237 238 308 310
snrfact	036720 automatic	float bin(27)	dcl 24 set ref 213 214
speed	032767 automatic	float bin(27)	dcl 4 set ref 229 231 235 257 289 296 304 354 359 363 371 418
speedb1	011710 automatic	float bin(27)	array dcl 2 set ref 454 540 545
speedb2	014252 automatic	float bin(27)	array dcl 2 set ref 455 5[illegible] 545
sysin	000030 constant	file	input stream dcl 34 set ref 34 55 61 64 67 70

			74 86 89 92 115 569 572 577 581 599
sysprint	000032 constant	file	output stream dcl 34 set ref 34 48 49 54 60 63 66 69 73 84 88 91 96 97 114 121 125 194 197 288 289 295 296 354 359 363 460 462 464 469 471 473 475 477 481 483 485 490 492 493 501 502 504 508 514 517 522 527 532 537 542 547 552 557 562 593 597 622
t	035363 automatic	float bin(27)	array dcl 20 set ref 43 231
tbarge	033004 automatic	float bin(27)	dcl 4 set ref 187 188 208 209 210 215 216 219 219 231 390 437
tdwt	032771 automatic	float bin(27)	dcl 4 set ref 316 320 320 322 323
temp	033031 automatic	float bin(27)	dcl 4 set ref 188 192 263 285 288
tn	036637 automatic	float bin(27)	dcl 21 set ref 231
tlink	033022 automatic	float bin(27)	dcl 4 set ref 107 115 257 473
totcapcost	033303 automatic	float bin(27)	dcl 4 set ref 344 349 352
totopcost	033302 automatic	float bin(27)	dcl 4 set ref 345 350 352 376 423
tport1	033077 automatic	float bin(27)	dcl 4 set ref 249 251 252 266 274 275 280 293 305
tport2	033100 automatic	float bin(27)	dcl 4 set ref 253 255 256 266 275 279 280 293 305
ttript	033106 automatic	float bin(27)	dcl 4 set ref 258 259 263 283 284 285 288 306 307 315 363 394 441
tugopdays	033020 automatic	float bin(27)	dcl 4 set ref 107 115 315 469
tunlink	033023 automatic	float bin(27)	dcl 4 set ref 107 115 257 475
twait1	033024 automatic	float bin(27)	dcl 4 set ref 261 266 267 274 280 283 293 299 359 396
twait2	033025 automatic	float bin(27)	dcl 4 set ref 261 267 275 279 283 293 299 397
vcargo	033112 automatic	float bin(27)	dcl 4 set ref 107 115 325 481
volwt	036723 automatic	float bin(27)	dcl 24 set ref 216
w	035724 automatic	float bin(27)	array dcl 20 set ref 41 231
wa	036636 automatic	float bin(27)	dcl 21 set ref 231
wmacht	033044 automatic	float bin(27)	dcl 4 set ref 242
wmisc	033045 automatic	float bin(27)	dcl 4 set ref 107 115 316
woutfitb	036647 automatic	float bin(27)	dcl 22 set ref 189 190 201 216
woutfitt	033043 automatic	float bin(27)	dcl 4 set ref 241 244
wsteelb	036646 automatic	float bin(27)	dcl 22 set ref 184 190 201 216
wsteelt	033042 automatic	float bin(27)	dcl 4 set ref 240 244
x	000100 automatic	float bin(27)	array dcl 588 set ref 605 609 610 615 619 620
x1cb	036632 automatic	float bin(27)	dcl 21 set ref 172 180 184
x2cb	036633 automatic	float bin(27)	dcl 21 set ref 173 180 184
x3cb	036634 automatic	float bin(27)	dcl 21 set ref 174 180 184
xcb	036631 automatic	float bin(27)	dcl 21 set ref 171 172 172 173 173 174 174
xmax	001457 automatic	float bin(27)	dcl 589 set ref 596 599 600
xmin	001456 automatic	float bin(27)	dcl 589 set ref 596 599 600
y1	000472 automatic	float bin(27)	array dcl 588 set ref 606 609 616 619
y2	001064 automatic	float bin(27)	array dcl 588 set ref 607 610 617 620
ymax	001461 automatic	float bin(27)	dcl 589 set ref 596 599 600
ymin	001460 automatic	float bin(27)	dcl 589 set ref 596 599 600

```
NAMES DECLARED BY EXPLICIT CONTEXT.
drop_and_swap               003434 constant    entry            external dcl 1 ref 1
dslp                        010255 constant    label            dcl 258 ref 258 406 412
dwtlp                       005732 constant    label            dcl 165 ref 165
endbtlp                     013420 constant    label            dcl 449 ref 219 233 449
endsblp                     013430 constant    label            dcl 451 ref 199 451
endlblp                     013424 constant    label            dcl 450 ref 227 450
endwtlp                     013434 constant    label            dcl 452 ref 452
format1                     003361 constant    format           ref 460 462
format2                     003304 constant    format           ref 469 471 473 475 477
format3                     003215 constant    format           ref 483 485
itolp                       010574 constant    label            dcl 302 ref 292 297 302 404 414
lloop                       007322 constant    label            dcl 195 ref 192 195
next3                       010513 constant    label            dcl 293 ref 285 293
next4                       010625 constant    label            dcl 308 ref 300 308
next5                       013441 constant    label            dcl 453 ref 453
plots                       022002 constant    entry            internal dcl 585 ref 520 525 530 535 540 545 550
                                                                555 560 564 585
sp                          004320 constant    label            dcl 63 ref 58 63
st                          004147 constant    label            dcl 48 ref 48 516 566
st2                         004532 constant    label            dcl 84 ref 84 99
st3                         005331 constant    label            dcl 130 ref 103 130
stgr                        022040 constant    label            dcl 591 ref 591 624
yesno                       021627 constant    entry            internal dcl 567 ref 51 122 126 515 519 524 529
                                                                534 539 544 549 554 559 563 567 594 623

NAMES DECLARED BY CONTEXT OR IMPLICATION.
abs                                            builtin function internal ref 192 285
ceil                                           builtin function internal ref 155 157 158 159 161 162 204 259 284
                                                                307
exp                                            builtin function internal ref 177 178 179 181 182 183
max                                            builtin function internal ref 146 189 259 266 266 275 280 284 307
                                                                334 337 339
min                                            builtin function internal ref 146

STORAGE REQUIREMENTS FOR THIS PROGRAM.

          Object     Text      Link      Symbol    Defs      Static
Start          0        0     23132       23206   22726       23142
Length     24410    22726        54        1166     203           6

BLOCK NAME                  STACK SIZE     TYPE                WHY NONQUICK/WHO SHARES STACK FRAME
drop_and_swap                    16496 external procedure  is an external procedure.
yesno                              236 internal procedure  contains a format statement, and uses I/O statements.
plots                             1030 internal procedure  contains a format statement, and uses I/O statements.

THE FOLLOWING EXTERNAL OPERATORS ARE USED BY THIS PROGRAM.
fx1_to_fl2          r_l_a               r_g_a               r_e_as              r_ge_a              call_ext_out_desc
call_ext_out        call_int_this_desc  call_int_this       call_int_other      return              fl2_to_fx1
ext_entry           int_entry           int_entry_desc      get_end             put_end             stream_io
open                close               trunc_fx2           ceil_fl             put_data_eis        put_list_eis
get_edit_eis        get_list_eis        real_to_real_rd     any_to_any_rd       divide_fx1          exp
dbl_p_dbl           real_p_real         put_field           put_field_chk       put_control
```

A.3 DICTIONARY OF VARIABLES USED IN THE DROP-AND-SWAP COMPUTER MODEL

DEFINITION OF VARIABLES IN THE PROGRAM "drop-and-swap"

Variable	Definition (units)
acargo:	Annual Time Value for Cargo Cost. Total annual cost for the time value of the cargoes onboard the barge while being transported. The product of the annual discount rate times the seatime times the cargo value. ($)
acost:	Annual Cost. Total annual cost for either drop-and-swap or integral OGTB system, including annual operating costs and annualized present value share of capital costs. ($)
acrew:	Annual Crew Cost. Annual costs for crew, including wages and subsistence, for a tug. ($/tug)
admin:	Annual Administrative Costs. Annual costs for adminstration of tug. ($/tug)
aflowave1:	Annual Average Cargo Flow from Port 1. Average annual flow of cargo to be loaded at Port 1 and discharged at Port 2. (cargo units)
aflowave2:	Annual Average Cargo Flow from Port 2. Average annual flow of cargo to be loaded at Port 2 and discharged at Port 1. (cargo units)
aflowcap:	Annual Flow Capacity. Annual amount of cargo capacity provided by each tug, product of barge DWT and number of tug voyages. (cargo units)
afuel:	Annual Fuel Cost. Annual cost for fuel for a tug; product of the number of tug voyages, fuel consumption per voyage, and unit cost of fuel. ($/tug)
ainsur:	Annual Insurance Premiums. Annual cost for insurance for a tug-barge unit. ($/tug-barge)

alube: Annual Lube Oil Costs. Annual cost for lube oil for a tug; product of the number of tug voyages, lube oil consumption per voyage and unit cost of lube oil. ($/tug)

amandr: Annual M&R Costs. Annual cost for maintenance and repair of a tug-barge unit. ($/tug-barge)

aopcost: Annual Operating Cost. Annual costs for operating a tug-barge unit. ($/tug-barge)

aother: Annual Other Costs. Other annual miscellaneous operating costs not included in admin, afuel, alube, ainsur, amandr, acrew, asupplies, and aport. ($/tug)

aport: Annual Port Charges. Annual costs for port charges for each tug, including variable and fixed charges for Port 1 and Port 2. ($/tug)

astorcap: Annual Capital Cost for Storage Facilities. The cost for shoreside storage facilities required for the integral mode of operation. The product of the number of loading/discharging facilities, the barge size, and a cost factor ("cfixstor"). ($)

astorop: Annual Operating Costs for Storage Facilites. The annual operating cost incurred for the operation or use of shoreside storage facilities required for the integral mode of operation. The product of the annual cargo flows and a cost factor ("cvarstor"). ($/yr)

asupplies: Annual Supplies Cost. Annual costs for stores, supplies, and equipment for each tug-barge unit. ($/tug-barge)

atermcap: Total Capital Cost for Loading/Discharging Facilities. The capital cost for shoreside terminal facilities. The product of the terminal facilities throughput rate and a cost factor ("cfixterm"). ($)

atermop: Annual Operating Cost for Loading/Discharging Facilities. The annual operating costs incurred for the operation or use of the shoreside terminal facilities. The product of the annual cargo flows and a cost factor ("cvarterm"). ($/yr)

bargeopdays: Barge Operating Days. Number of days during the year which a barge is expected to be available for operations. (days)

bbarge: Barge Beam. (ft)

best1: Storage Matrix that saves the best values found for various drop-and-swap mode variables (specified below) for a specified port separation distance and port loading and discharge rate while iterating over tug speed, barge DWT, and barge form.

best2: Same as "best1" except for integral mode.

best1(1),	best2(1)	-	rfr
best1(2),	best2(2)	-	dwt/1000
best1(3),	best2(3)	-	speed
best1(4),	best2(4)	-	acost/1000000
best1(5),	best2(5)	-	ctug/1000000
best1(6),	best2(6)	-	cbarge/1000000
best1(7),	best2(7)	-	aopcost
best1(8),	best2(8)	-	totopcost
best1(9),	best2(9)	-	ctugbarges
best1(10),	best2(10)	-	alube
best1(11),	best2(11)	-	afuel
best1(12),	best2(12)	-	amandr
best1(13),	best2(13)	-	ainsur
best1(14),	best2(14)	-	asupplies
best1(15),	best2(15)	-	aport
best1(16),	best2(16)	-	resist
best1(17),	best2(17)	-	ihp
best1(18),	best2(18)	-	nrtrips
best1(19),	best2(19)	-	mintug, minitb
best1(20),	best2(20)	-	lbarge
best1(21),	best2(21)	-	bbarge
best1(22),	best2(22)	-	tbarge
best1(23),	best2(23)	-	displ
best1(24),	best2(24)	-	portimet/24
best1(25),	best2(25)	-	seatimet/24
best1(26),	best2(26)	-	ttript
best1(27),	best2(27)	-	aflowcap
best1(28)			twait1
	best2(28)	-	cb
best1(29)			twait2
	best2(29)	-	lb
best1(30)			minbarge1
	best2(30)	-	bt
best1(31)			minbarge2
	best2(31)	-	ld
best1(32)			cb

best1(33)	lb
best1(34)	bt
best1(35)	ld

blurb1,2,3: Subroutine Plots input parameters representing title of graph, vertical axis, and horizontal axis.

bt: Breadth-Draft Ratio. The ratio of the tug-barge unit's breadth and draft. (ft)

cbarge: Cost of Barge. ($)

c1,c2: Coefficient arrays used only as input parameters in the calling of subprogram "power"; values are used in subroutine "prop" of subprogram "power."

cb: Block Coefficient. The block coefficient of the tug-barge unit.

cf: Coefficient of Frictional Resistance for OGTB.

cfixport1: Fixed Port Charges for Port 1. Cost of fixed port charges of tug/barge at Port 1 per voyage. ($/voyage)

cfixport2: Fixed Port Charges for Port 2. Cost of fixed port charges of tug/barge at Port 2 per voyage. ($/voyage)

cfixstor: Storage Facility Capital Cost Factor. The cost per long ton for shoreside storage facilities used in the integral mode. ($/LT)

cfixterm: Terminal Facility Capital Cost Factor. The cost per long ton per day throughput rate for terminal facilities. ($/LT-day)

cfuel: Cost of Fuel. Diesel fuel cost per ton. ($/LT)

clube: Cost of Lube Oil. Lube oil cost per gallon. ($/Gal)

cmacht: Tug Machinery Cost. Cost of propulsion machinery onboard the tug; function of IHP. ($1000/tug)

countr: Index for the loadline coefficient array(llcoef) used in the calculation of the variable "minfbd".

countfitb: Barge Outfit Cost. Cost of a ton of outfit material onboard the barge. ($1000/LT)

coutfitt: Tug Outfit Cost. Cost of a ton of outfit onboard the tug. ($1000/LT)

csteelb: Barge Steel Cost. Cost of a ton of hull steel onboard the barge. ($1000/LT)

csteelt: Tug Steel Cost. Cost of a ton of hull steel onboard the tug. ($1000/LT)

csubs: Cost of Subsistence. Annual subsistence cost per crew member. ($/man)

ctug: Cost of Tug. Initial capital cost of a tug as a function of steel and outfit weight and machinery cost. ($/tug)

ctugbarges: Cost of Tug and Barges. Total capital cost of all tugs and barges in the system. ($)

cvarport1: Variable Port Charges for Port 1. Cost per cargo unit capacity for port charges at Port 1 per voyage. ($/cargo unit and voyage)

cvarport2: Variable Port Charges for Port 2. Cost per cargo unit capacity for port charges at Port 2 per voyage. ($/cargo unit and voyage)

cvarstor: Storage Facility Operating Cost Factor. The cost per long ton stored in the integral mode. ($/LT)

cvarterm: Terminal Facility Operating Cost Factor. The cost per long ton moved through a terminal facility. ($/LT)

cwage: Cost of Wages. Annual benefit and wage cost per crew member. ($/man)

d: Distance Index. Index used to represent the variable distance in optimum system arrays-rfrb1, rfrb2, dwtb1,...mintugs.

dbarge: Depth of Barge. (ft)

delay1: Port 1 Delay Time. Length of delay at Port 1 prior to and after cargo operations, including docking/undocking and awaiting berth time for the barge. (hrs)

delay2: Port 2 Delay Time. Same as "delay1" except for Port 2.

delbt: Breadth-Depth Ratio Increment. The incremental B/T to be used while varying the tug-barge breadth-depth ratio from its minimum to its maximum value.

delcb: Block Coefficient Increment. The incremental Cb to be used while varying the tug-barge block coefficient from its minimum to its maximum value.

deldist: Distance Increment. The incremental distance to be used while varying the port separation distance from its minimum to its maximum value. (nautical miles)

deldwt: Deadweight Increment. The incremental deadweight to be used while varying the barge deadweight from its minimum to its maximum value. (cargo units)

dellb: Length-Breadth Ratio Increment. The incremental lb to be used while varying the tug-barge length-breadth ratio from its minimum to its maximum value.

delrate: Loading/Discharge Rate Increment. The incremental L/D rate to be used while varying the L/D from its minimum to its maximum value. (cargo units/day)

delspeed: Tug Speed Increment. The incremental tug speed to be used while varying the tug speed from its minimum to its maximum value. (kts)

dhp: Delivered Horsepower. Horsepower required to be delivered to the propeller to drive the tug-barge unit "speed" knots.

displ: Barge Displacement. (LT)

disrate: Discount Rate. Discount rate to be used in present value calculations.

distance: Port Separation Distance. Distance between Port 1 and Port 2 in miles. (nautical miles)

dq: Coefficient array used only as an input parameter in the calling of subprogram "power"; values are used in subroutine "prop" of subprogram "power."

drange: Distance Range. Number of port separation distances to be investigated.

dsopt: Drop and Swap Option. This option enables the program to investigate two types of drop-and-swap configurations. When dsopt=0, both ports have a minimum of one barge stationed for L/D. When dsopt=1, the port with lowest L/D time has no barges stationed for L/D and tug remains with barge while in port.

dt: Coefficient array used only as an input parameter in the calling of subprogram "power"; values are used in subroutine "prop" of subprogram "power."

dwt: Barge Deadweight. (cargo capacity units)

dwtb1: Array variable used to store the barge deadweight size of the optimum drop-and-swap system for given port separation distance and L/D rates. (cargo capacity units)

dwtb2: Same as dwtb1 except for integral system. (cargo capacity units)

econlife: Economic Life. Economic life of the OTGB system used in present value calculations. (yrs)

ehp: Effective Horsepower. Horsepower required to be delivered by the propeller to drive the tug-barge unit "speed" knots.

eta: Relative Rotative Efficiency Array. Array of relative rotative efficiency (x1000) values used as input parameters in the calling of subprogram "power"; values are used in subroutine "propfactors" of subprogram "power."

fbd: Barge Freeboard. (in)

fuelcons: Fuel Consumption. Amount of fuel consumed by the tug per voyage. (LT/voyage)

fuelmargin: Fuel Margin. Percentage of fuel above "fuelcons" required to be onboard tug after bunkering.

graph_type: Subroutine Plots inputs parameter used to designate type of graphical output. If equals 0, loading rate is abscissa; if equals 1, distance is abscissa.

hr: Relative Rotative Efficiency. Relative rotative efficiency (η_R) of tug-barge hull form.

hullwt75: Hull steel weight for tug-barge unit with Cb=0.75 as a function of lb, bt, and dwt. (LT)

hullwt80: Hull steel weight for tug-barge unit with Cb=0.80 as a function of lb, bt, and dwt. (LT)

hullwt85: Hull steel weight for tug-barge unit with Cb=0.85 as a function of lb, bt, and dwt. (LT)

i: Index Parameter used in do loop to solve for Coefficient of Frictional Resistance for OGTB.

ibit: Subroutine "yesno" output value designating response of user to a question. If yes was signified, ibit equals 1; if no was signified, ibit equals 0.

ihp: Installed Horsepower. Horsepower of engines installed onboard tug. (horsepower)

index: Subprogram "power" Error Index. An index used to indicate if some calculation error occurred during the call of subprogram "Power".

inflafctr: Inflation Factor. A factor used to correct the total annual cost ("acost") for inflation after January 1979.

inrate: Loading/Discharging Rate Input Option Parameter. If equals 0, then the rate of loading and discharge at each port will be the same and equal at each port and will be systematically varied from the minimum to the maximum specified value. If equals 1, the loading and discharge rate at each port will be required input.

iopt: Drop and Swap/Integral Mode Option. This option is used in the main computations. When it equals 0, then drop-and-swap calculations are executed. When it equals 1, then integral mode calculations are executed.

iopt3: Detailed Output Option. This option is used to specify how much detail is desired in the output. When it equals 0, only the optimum drop-and-swap and integral system values are printed out for a given distance and loading/discharging rate. When it equals 1, drop-and-swap and integral mode

values are printed for every value of dwt, speed, distance, rate, cb, bt, and lb.

iout: Output of Input Data Option. This option is used to insure that the input parameter values are printed only once. When 0, input is printed; when 1, input is not printed.

iout2: Output of Nonconvergence Message Option. This option is used to insure that the message of nonconvergence of the waiting time routine is printed only once for a given L/D rate and port separation distance. If equals 0, message will be printed. When equals 1, message will not be printed.

iout3: Form Coefficient Under/Overflow Indicator. This variable is set to 1 if either lb, bt, or cb values are out of interpolable ranges. Otherwise its value is zero. It is used in the program to allow the user to revise his form coefficient inputs without having to input any other data.

j: Index parameter used in do loop to solve for waiting times, voyage times, and minimum tugs required in drop-and-swap mode. Also used in Yesno Subroutine.

lb: Lenth-Breadth Ratio. The ratio of tug-barge unit's length and breadth. (ft)

lbarge: Length of Barge. (ft)

lbarge75: Length of barge with Cb=0.75 as a function of lb, bt, and dwt. (ft)

lbarge80: Length of barge with Cb=0.80 as a function of lb, bt, and dwt. (ft)

lbarge85: Length of barge with Cb=0.85 as a function of lb, bt, and dwt. (ft)

ld: Length-Depth Ratio. The ratio of barge length and barge draft. (ft)

ldfact: Length-Depth Ratio Loadline Factor. This factor is used to correct the tabulated minimum freeboard value (minfbd) for length-depth ratios less than 15.

litb: Length of Integrated Tug/Barge Combination. (ft)

llcoef: Minimum Loadline Coefficient Array. From this array the value of minfbd is found via linear interpolation with respect to length.

llfact: Minimum Loadline Coefficient File. The file in which llcoef is stored.

ltug: Length of Tug. (ft)

lubecons: Lube Oil Consumption. Amount of lube oil consumed by the tug per voyage. (gals/voyage)

maxb: Maximum Beam. Maximum barge beam allowed during voyage. (ft)

maxbt: Maximum Breadth-Draft Ratio. Maximum tug-barge bt value to be investigated.

maxcb: Maximum Block Coefficient. Maximum barge block coefficient to be investigated.

maxdist: Maximum Distance. Maximum port separation distance to be investigated. (nautical miles)

maxdwt: Maximum Deadweight. Maximum barge cargo deadweight capacity to be investigated. (cargo units)

maxl: Maximum Length. Maximum OGTB length allowed during voyage. (ft)

maxlb: Maximum Length-Breadth ratio. Maximum tug-barge L/B value to be investigated.

maxrate: Maximum Loading/Discharge Rate. Maximum port L/D rate to be investigated. (cargo units/day)

maxspeed: Maximum Speed. Maximum tug speed to be investigated. (kts)

maxt1: Maximum Draft Port 1. Maximum allowed draft in Port 1. (ft)

maxt2: Maximum Draft Port 2. Maximum allowed draft in Port 2. (ft)

mflowave1: Monthly Average Cargo Flow from Port 1. Average monthly flow of cargo to be loaded at Port 1 and discharged at Port 2. (cargo units)

mflowave2: Monthly Average Cargo Flow from Port 2. Average monthly flow of cargo to be loaded at Port 2 and discharged at Port 1. (cargo units)

minbarge1: Minimum Barges/Terminal Facilities at Port 1. Minimum number of terminal facilities (and barges for drop-and-swap mode only) required at Port 1 in order to be compatible with Port 1 loading/discharging rates and flow requirements. (barges/facilities)

minbarge2: Same as minbargel except for Port 2. (barges/facilities)

minbarges1: Storage array for "minbarge1" for the port pair trades under consideration. (barges/facilities)

minbarges2: Same as minbarges1 except for Port 2. (barges/facilities)

minbrg1,2: Storage variables for values of minbarge1 and minbarge2, respectively, used for drop-and-swap calculations when dsopt is 0.

minbt: Minimum Breadth-Draft Ratio. Minimum tug-barge B/T value to be investigated.

mincb: Minimum Block Coefficient. Minimum tug-barge Cb value to be investigated.

mindist: Minimum Distance. Minimum port separation distance to be investigated. (nautical miles)

mindwt: Minimum Deadweight. Minimum barge cargo deadweight capacity to be investigated. (cargo units)

mindwttemp: Temporary storage variable for the variable "mindwt" used for reducing the number of "dwt" iterations required.

minfbd: Minimum Freeboard. The uncorrected (or block coefficient corrected) minimum freeboard value as obtained after linear interpolation of the array llcoef.

minitb: Minimum OGTB. The minimum number of OGTB's required in the integral mode to provide sufficient flow capacity. (tug-barges)

minlb: Minimum Length-Breadth ratio. Minimum tug-barge

L/B value to be investigated.

minrate: Minimum Loading/Discharge Rate. Minimum port L/D rate to be investigated. (cargo units/day)

minspeed: Minuimum Speed. Minimum tug operating speed to be investigated. (kts)

mintug: Minimum Tugs. Minimum number of tugs required in the drop-and-swap mode to provide sufficient flow capacity. (tugs)

mintugs1: Storage array for the minimum number of tugs required by the optimal drop-and-swap system for a given L/D rate and port separation distance. (tugs)

mintugs2: Storage array for the minimum number of OGTB's required by the optimal integral system for a given port L/D rate and port separation distance. (OGTBs)

multifctr1-3: Multiple Ship Cost Reduction Factors. Factors used to correct single unit costs for multi-unit orders for tugs and barges.

nextchar: Yesno Subroutine input variables used to test if yes or no response was given by user.

nrcrew: Number of Crew. Number of crew members required for a tug. (men)

nrtrips: Number of Trips. Number of voyages tug or OGTB can make during a year. (voyages/year)

o1,o2: Subroutine Plots Input Parameters representing the arrays of obscissa and ordinate coordinates of the points to be graphed.

output_type: Output Type Option. Allows user to designate whether printed output is desired. If equals 1, printed output is provided. If equals 0, printed output is not provided.

parameters: Name of file containing the semifixed parametric values used by the program.

percent: Array used to store the percentage that port time takes of the total voyage time for the port pair trades under consideration.

plot_: Subroutine used to graph output data.

plot_scale: Subroutine used to set abscissa and ordinate scale minimum and maximum values.

plot_setup: Subroutine used to set up the proper environment for the plot_ subroutine.

portimet: Tug Porttime. Time tug or OGTB remains in port during a voyage. (hrs)

power: Subprogram used to calculate the "dhp" and "ehp" as a function of the tug-barge system's principal dimensions and speed.

powerdata: File used to store data required by subprogram "power".

propef: Open Water Propeller Efficiency (η_o). The ratio of the power delivered by the propeller and the power delivered to the propeller in open water.

pvf: Present Value Factor. Factor used in present value calculations; a function of the discount rate and economic life for the system.

r: Iteration index used for port loading/discharging rates.

rate: Loading/Discharging Rates. Daily amount of cargo which can be either loaded into or discharged out of a barge located at either port. Refers to each barge of the minimum required at that port. (cargo units/day)

rc: Residual Resistance Array. Array of residual resistance coefficients used as input parameter for subprogram "power"; values are used in subroutine "resist" of subprogram "power."

resist: Resistance. Total still water hydrodynamic resistance of the OGTB combination.

rfr: Required Freight Rate. The amount of freight per cargo unit carried required to cover system operating and capital costs (present value annualized). ($/cargo unit)

rfrb1: Storage array for the minimum required freight rate found for a drop-and-swap system with

specified port L/D rate and port separation distance. ($/cargo unit)

rfrb2: Same as for rfrb1 except for integral mode. ($/cargo unit)

rload1: Loading Rate at Port 1. Daily cargo loading rate at Port 1. (cargo units/day)

rload2: Loading Rate at Port 2. Daily cargo loading rate at Port 2. (cargo/units/day)

rlubeoil: Lube Oil Consumption Rate. Amount of lube oil consumed per hour. (gal/hr)

rportfuel: Inport Fuel Consumption Rate. Amount of Diesel fuel consumed per hour while tug is in port. (ton/hr)

rrange: Rate Range. Number of L/D rates to be investigated.

rseafuel: At Sea Fuel Consumption Rate. Amount of diesel fuel to be consumed per hour by the tug while steaming. (tons/hr)

runload1: Unloading Rate at Port 1. Daily cargo discharge rate per terminal at Port 1. (cargo units/day)

runload2: Unloading Rate at Port 2. Daily cargo discharge rate per terminal at Port 2. (cargo units/day)

seatimet: Tug Seatime. Time tug or OGTB is at sea; including delay time and linkage time (drop-and-swap mode only). (hours per voyage)

servmargin: Service Margin. Engine horsepower service margin used to calculate IHP.

sfc: Specific Fuel Consumption. Amount of diesel fuel consumed per horsepower per hour by the tug at sea. (lbs/hp-hr)

shp: Shaft Horsepower. Shaft horsepower required by tug to push OGTB through the water at the specified speed. (horsepower)

shrfact: Sheer Correction Factor. Correction to the freeboard calculation due to lack of sheer expected in the barge form.

speed: Tug Speed. Tug designed service speed. (kts)

speedb1: Storage array for the speed of the drop-and-swap system for a given port L/D rate and port separation distance. (kts)

speedb2: Same as speedb1 except for integral mode. (kts)

sysin: System input file

sysprint: System output file

t: Thrust Deduction Fraction Array. Array of one minus the thrust deduction fraction (x1000) values used as input parameter in the calling of subprogram "power"; values are used in subroutine "propfactors" of subprogram "power."

tbarge: Barge Draft. (ft)

tdwt: Total Deadweight of OGTB. Includes cargo deadweight capacity plus fuel and miscellaneous weights. (cargo units)

temp: Temporary storage location used for various intermediate calculations.

th: Thrust Deduction Fraction. Thrust deduction fraction (t) of the tug-barge form.

tlink: Linking Time. Time required to link a tug with a barge for drop-and-swap operations. (hrs)

totcapcost: Total Capital Cost. Total system capital costs for tugs, barges, terminal facilities, (and storage facilities for integral mode only). ($)

totopcost: Total Operating Costs. Total OGTB system operating costs. For drop-and-swap mode it includes M&R and insurance costs for those barges in excess of the number of tugs. ($)

tport1: L/D Time Spent in Port 1. Amount of time required for L/D operations at Port 1 per barge. (hrs)

tport2: L/D Time Spent in Port 2. Amount of time required for L/D operations at Port 2 per barge (hrs)

ttript: Tug Total Voyage Time. Amount of time required

by tug for a voyage. (days)

tugopdays: Tug Operating Days. Number of days during the year in which a tug is expected to be available for operations. (days)

tunlink: Unlinking Time. Time required to disconnect barge from tug for drop-and-swap operations. (hrs)

twait1: Port 1 Waiting Time. Time tug is required to wait for barge at Port 1 during drop-and-swap operations. (hrs)

twait2 Port 2 Waiting Time. Time tug is required to wait for barge at Port 2 during drop-and-swap operations. (hrs)

vcargo: Cargo Value. Value of cargo per long ton. ($/LT)

volwt: Cargo Cubic Capacity. The amount of cargo that could fit within the cubic volume of the barge. (LT)

w: Wake Fraction Array. Array of one minus the wake fraction (x1000) values used as an input parameter in the calling of subprogram "power"; values are used in subroutine "propfactors" of subprogram "power."

wa: Wake Fraction. Wake fraction (W_T) of the tug-barge form.

wmacht: Tug Machinery Weight. (LT)

wmisc: Tug Miscellaneous Weight. (LT)

woutfitb: Barge Outfit Weight. (LT)

woutfitt: Tug Outfit Weight. (LT)

wsteelb: Barge Hull Steel Weight. (LT)

wsteelt: Tug Steel Weight. (LT)

x: Subroutine Plots variable used to represent abscissa array values.

xcb: Quadratic Interpolation Coefficient. Parameter used in the quadratic interpolation with respect

to block coefficient for the value of lbarge and wsteelb.

x1cb: Same as for xcb.

x2cb: Same as for xcb.

x3cb: Same as for xcb.

xmax,xmin: Subroutine Plots variables used to represent minimum and maximum range of abscissa axis.

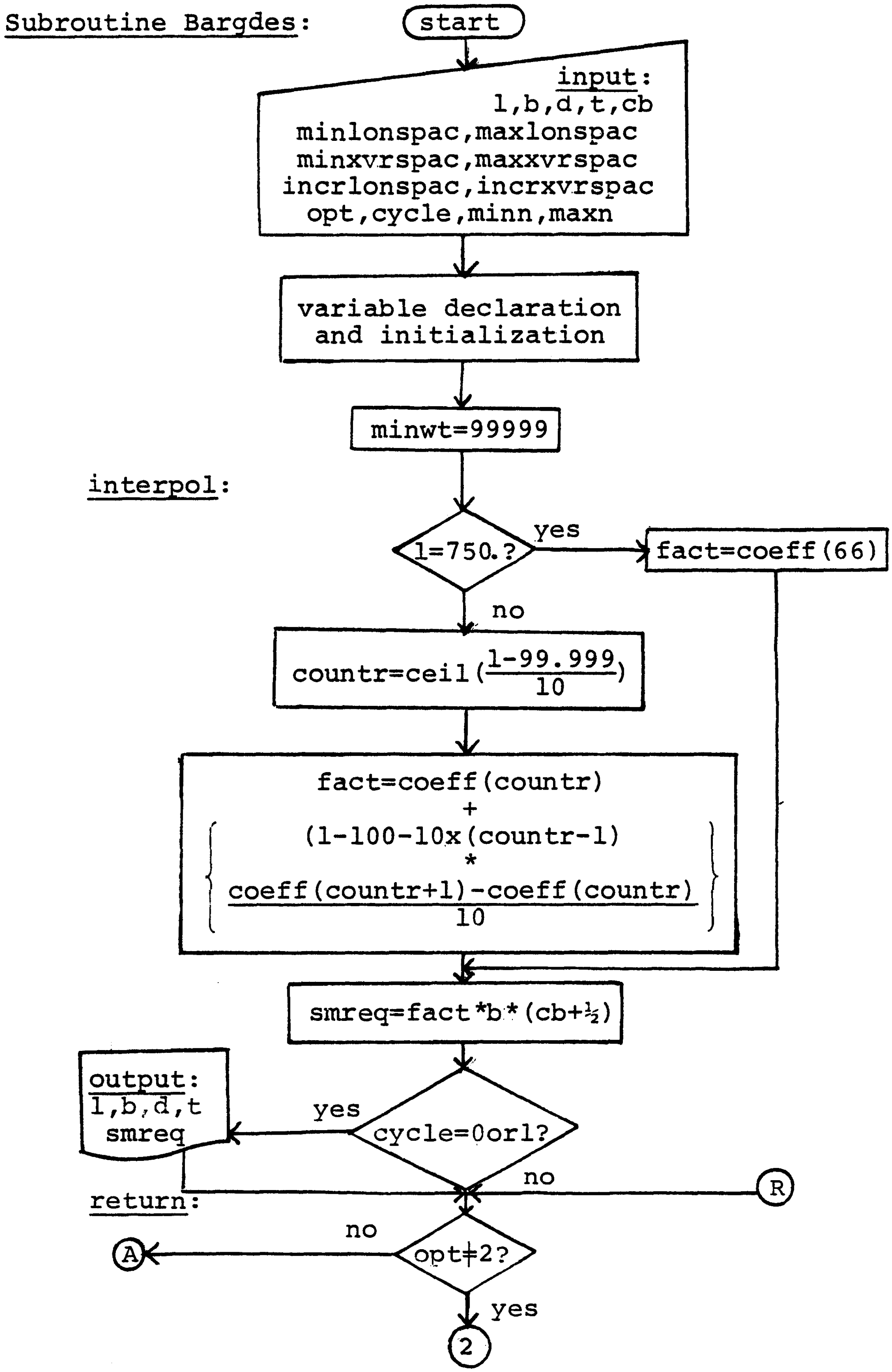
y1,y2: Subroutine Plots variables used to represent ordinate array values for drop-and-swap and integral modes respectively.

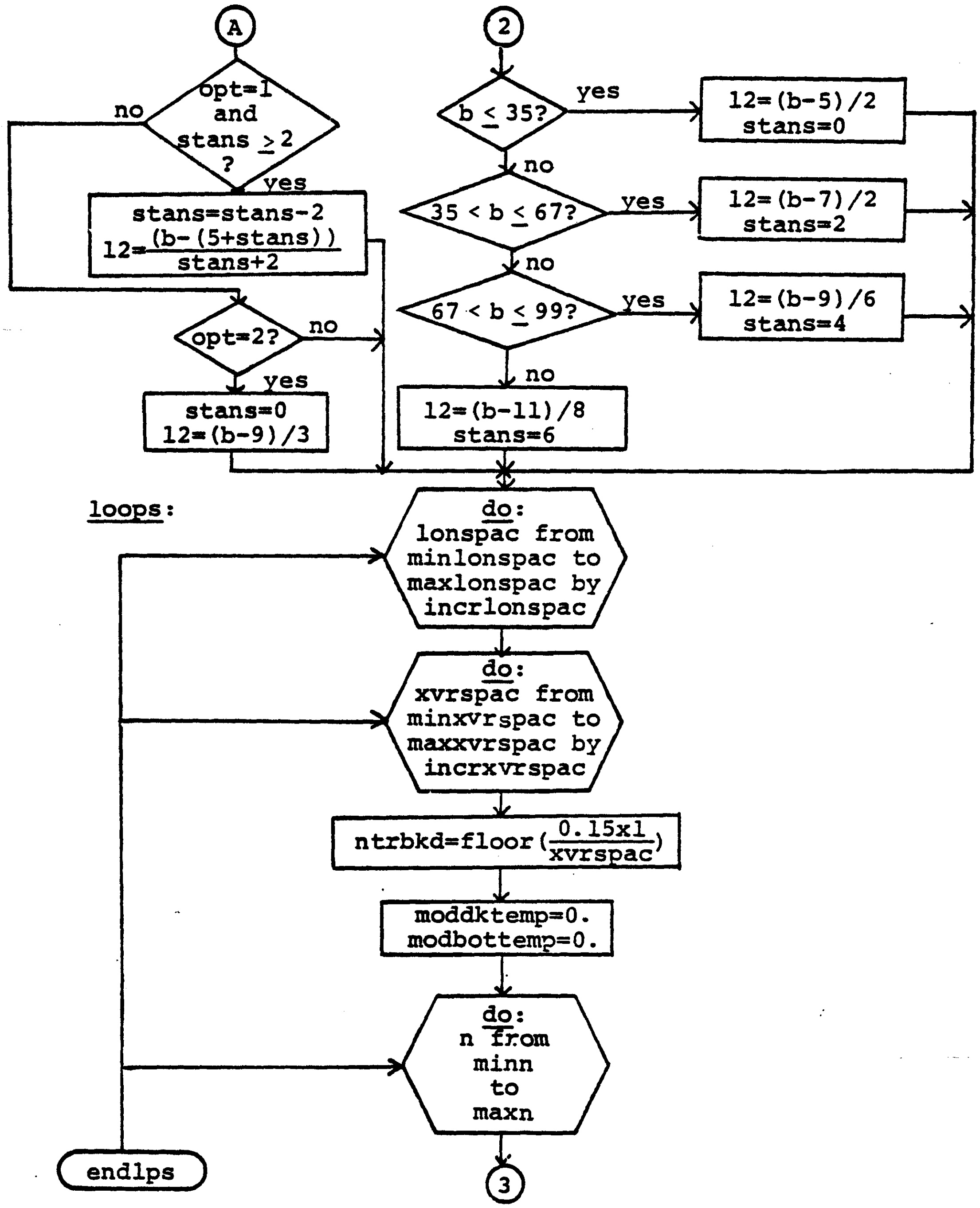
ymax,ymin: Subroutine Plots variables used to represent minimum and maximum range of ordinate axis.

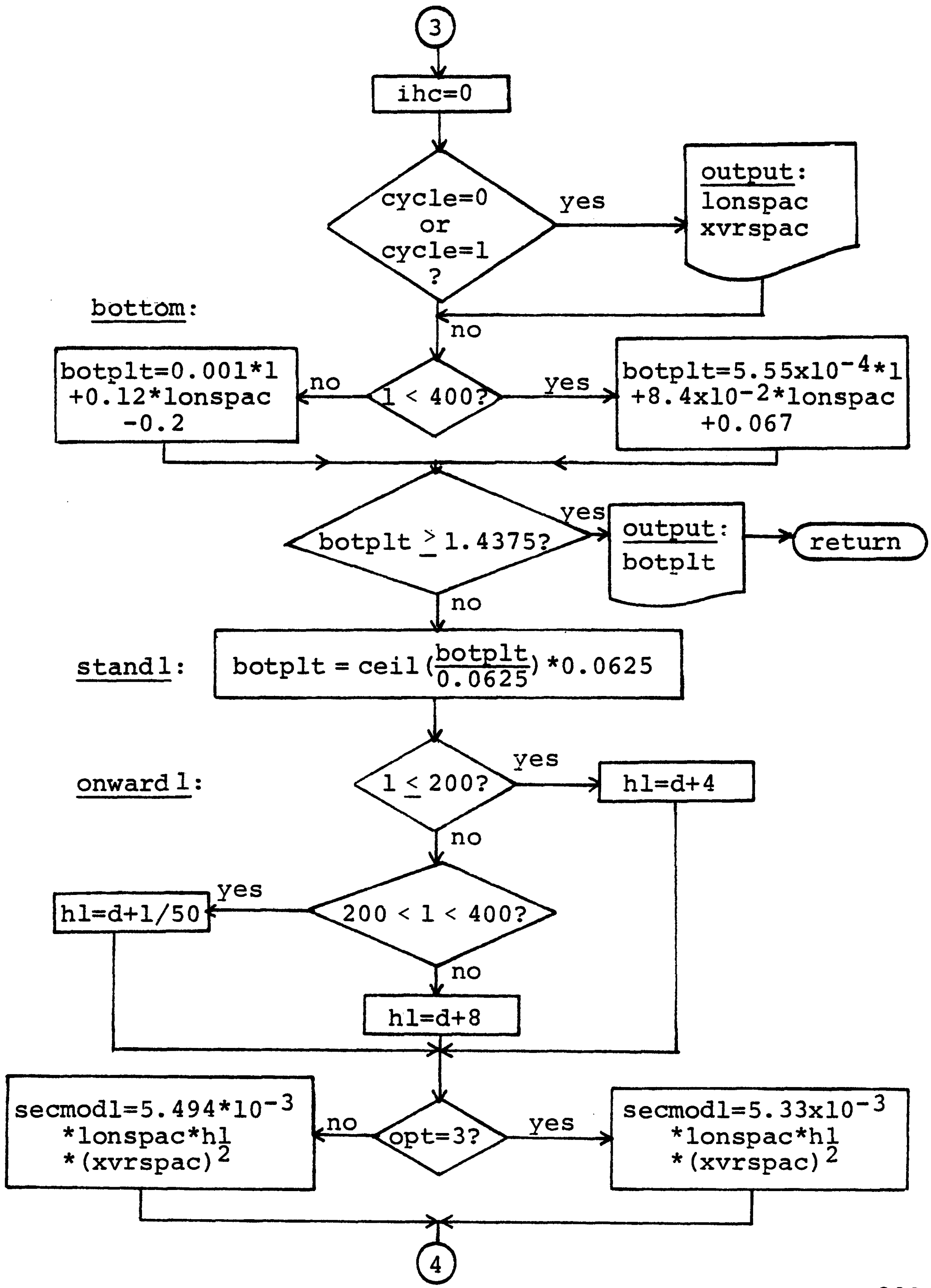
APPENDIX B

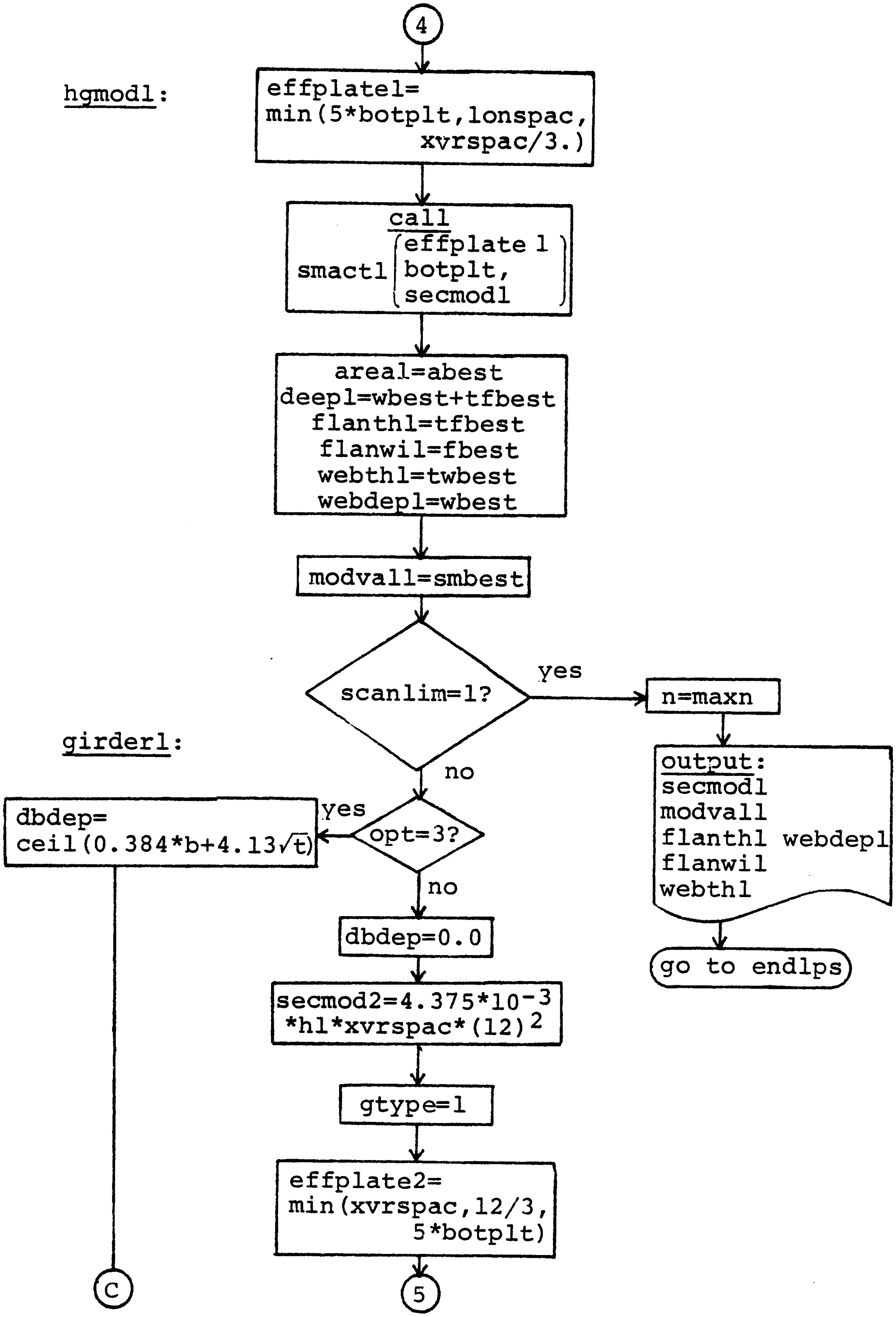
DOCUMENTATION FOR THE BARGE DESIGN MODEL

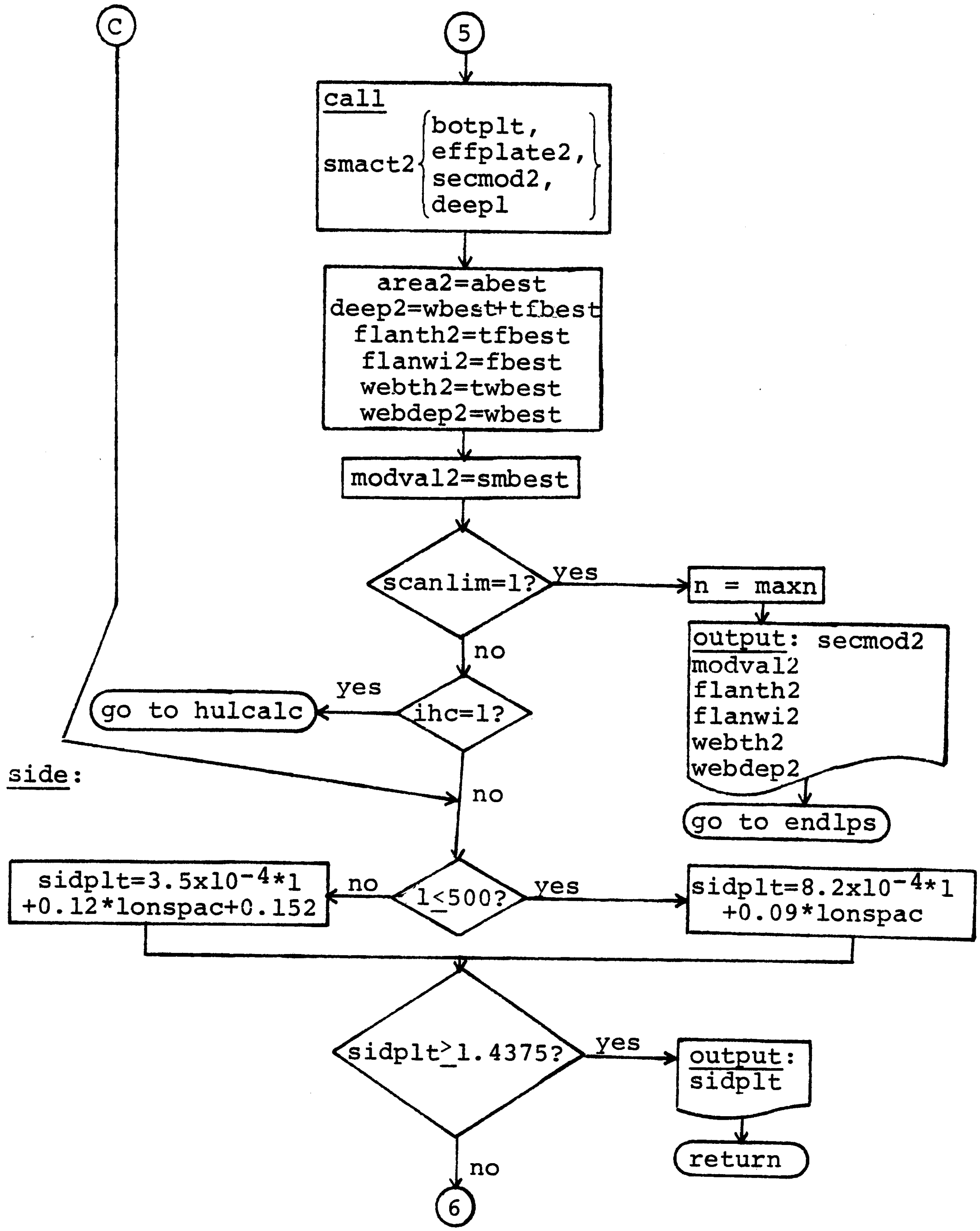
B.1 FLOWCHART OF THE BARGE DESIGN MODEL




A
opt=1 and stans ≥ 2 ?
no
yes
stans=stans-2
12=(b-(5+stans))/(stans+2)
opt=2?
no
yes
stans=0
12=(b-9)/3
2
b ≤ 35?
yes
12=(b-5)/2
stans=0
no
35 < b ≤ 67?
yes
12=(b-7)/2
stans=2
no
67 < b ≤ 99?
yes
12=(b-9)/6
stans=4
no
12=(b-11)/8
stans=6
loops:
do:
lonspac from minlonspac to maxlonspac by incrlonspac
do:
xvrspac from minxvrspac to maxxvrspac by incrxvrspac
ntrbkd=floor(0.15xl/xvrspac)
moddktemp=0.
modbottemp=0.
do:
n from minn to maxn
endlps
3

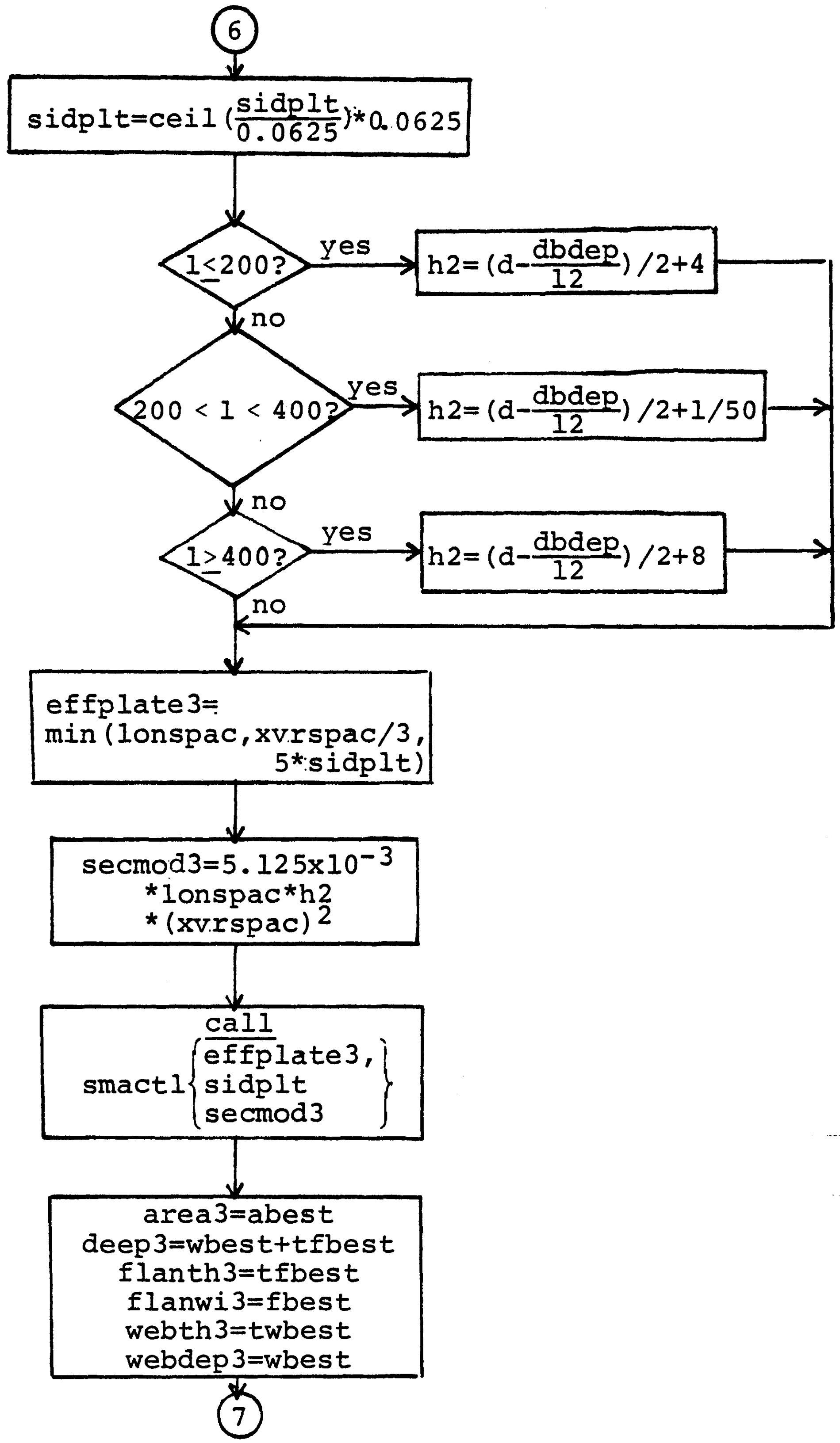

3
ihc=0
cycle=0 or cycle=1 ?
yes
output: lonspac xvrspac
bottom:
no
botplt=0.001*l +0.12*lonspac -0.2
no
l < 400?
yes
botplt=5.55x10-4*l +8.4x10-2*lonspac +0.067
botplt ≥ 1.4375?
yes
output: botplt
return
no
stand1:
botplt = ceil (botplt / 0.0625) *0.0625
onward1:
l ≤ 200?
yes
hl=d+4
no
yes
hl=d+l/50
200 < l < 400?
no
hl=d+8
secmodl=5.494*10-3 *lonspac*hl *(xvrspac)2
no
opt=3?
yes
secmodl=5.33x10-3 *lonspac*hl *(xvrspac)2
4


4
hgmodl:
effplatel=
min(5*botplt,lonspac,
xvrspac/3.)
call
smactl
effplate 1
botplt,
secmodl
areal=abest
deepl=wbest+tfbest
flanthl=tfbest
flanwil=fbest
webthl=twbest
webdepl=wbest
modvall=smbest
scanlim=1?
yes
n=maxn
output:
secmodl
modvall
flanthl webdepl
flanwil
webthl
go to endlps
no
girderl:
opt=3?
yes
dbdep=
ceil(0.384*b+4.13√t)
no
dbdep=0.0
secmod2=4.375*10-3
*hl*xvrspac*(12)2
gtype=1
effplate2=
min(xvrspac,12/3,
5*botplt)
C
5

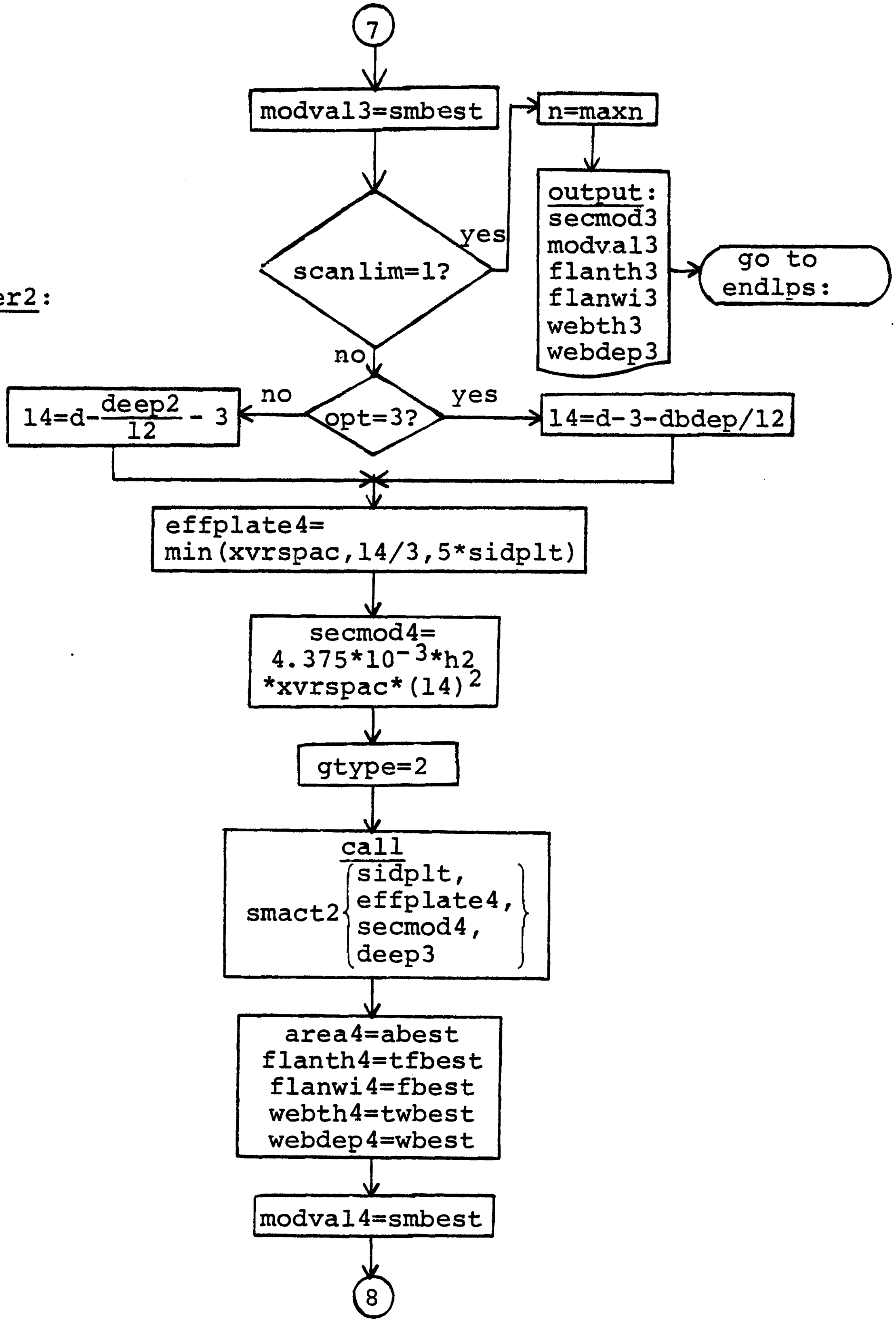

C
5
call
smact2
botplt,
effplate2,
secmod2,
deepl
area2=abest
deep2=wbest+tfbest
flanth2=tfbest
flanwi2=fbest
webth2=twbest
webdep2=wbest
modval2=smbest
scanlim=1?
yes
n = maxn
output: secmod2
modval2
flanth2
flanwi2
webth2
webdep2
go to endlps
no
ihc=1?
yes
go to hulcalc
no
side:
sidplt=3.5x10-4*l
+0.12*lonspac+0.152
no
l≤500?
yes
sidplt=8.2x10-4*l
+0.09*lonspac
sidplt≥1.4375?
yes
output:
sidplt
return
no
6

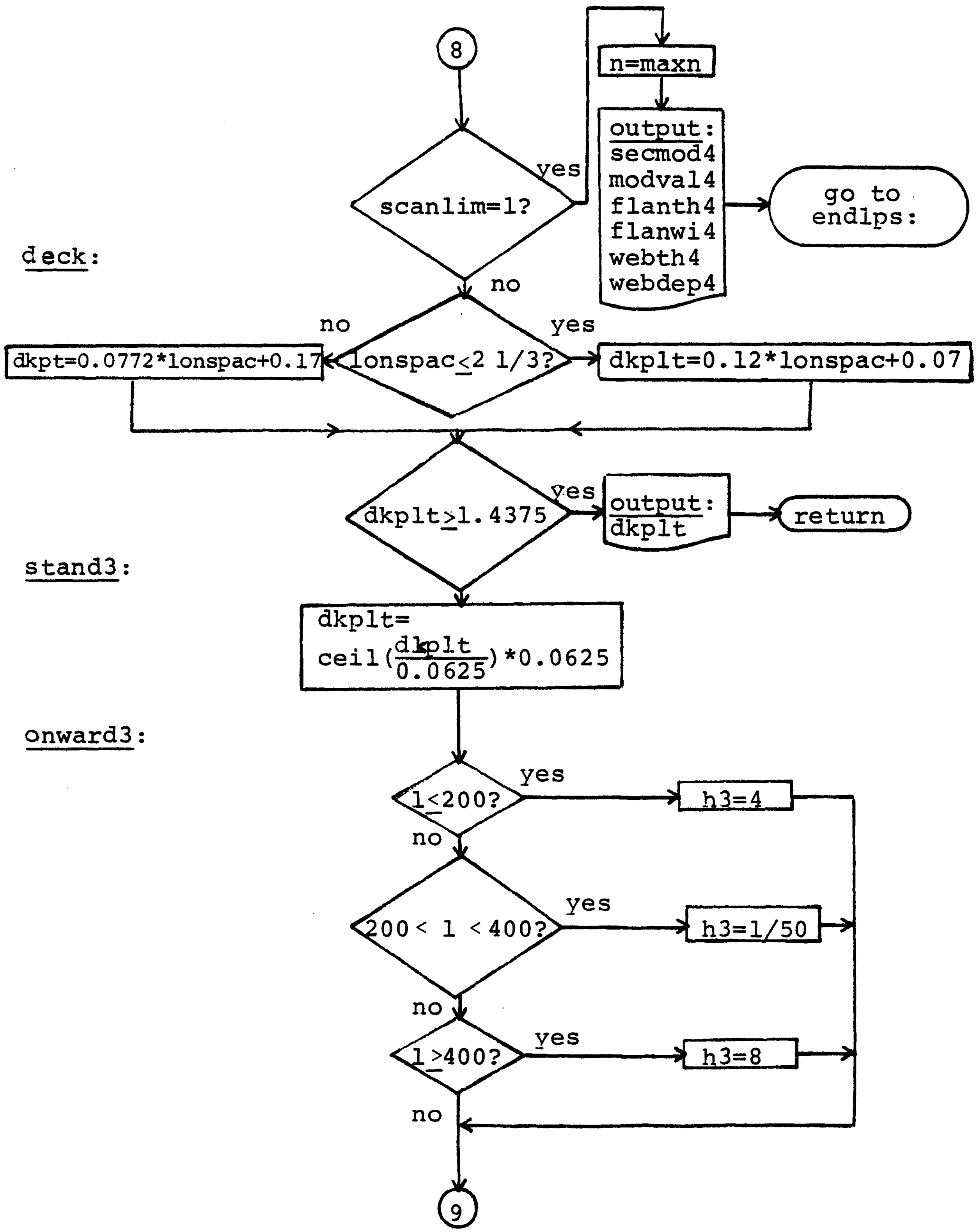
stand2:

onward2:



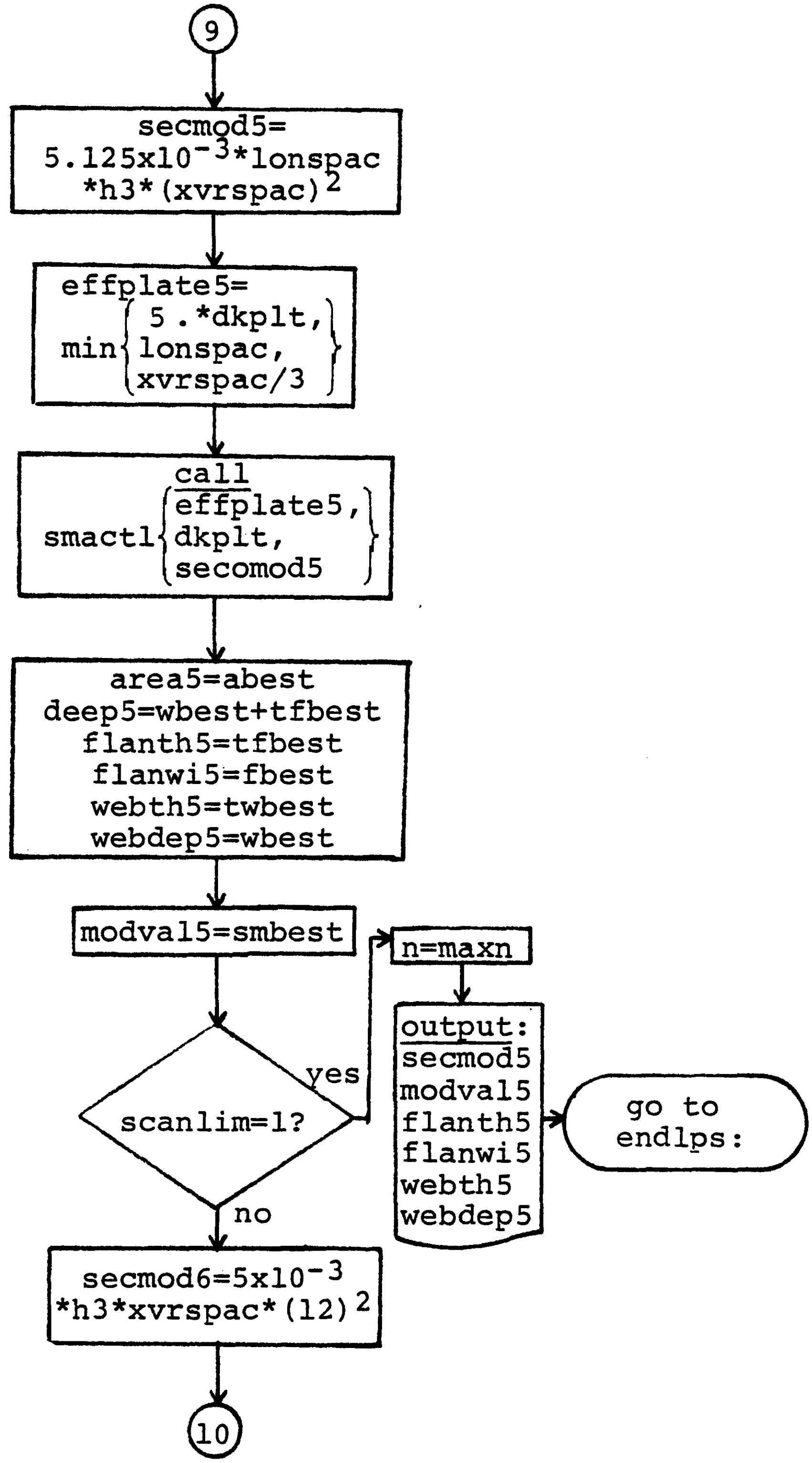
girder2:

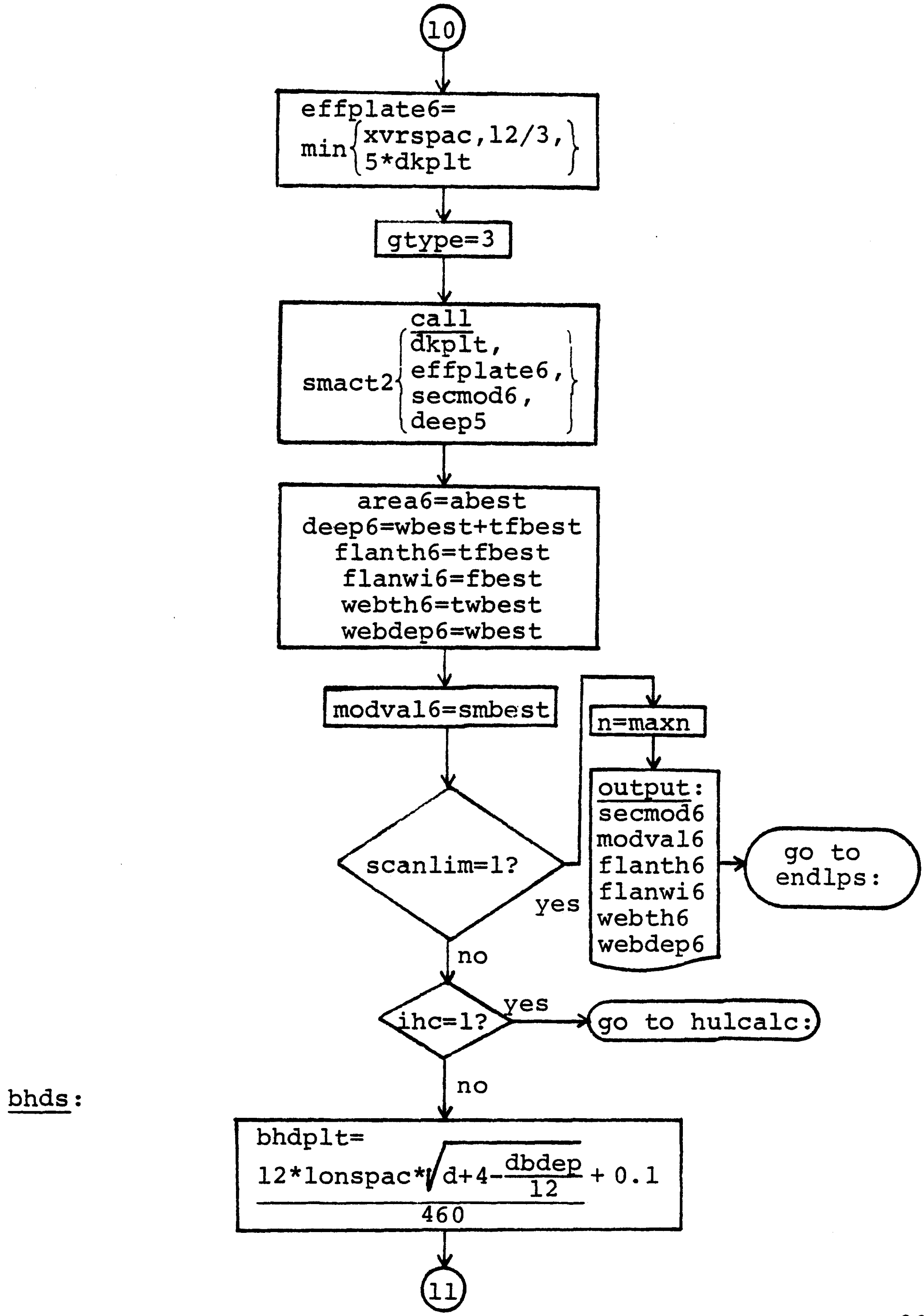

7
modval3=smbest
scanlim=1?
yes
n=maxn
output:
secmod3
modval3
flanth3
flanwi3
webth3
webdep3
go to endlps:
no
opt=3?
no
l4=d-deep2/12 - 3
yes
l4=d-3-dbdep/12
effplate4=
min(xvrspac,l4/3,5*sidplt)
secmod4=
4.375*10^-3*h2
xvrspac(l4)^2
gtype=2
call
smact2{sidplt,
effplate4,
secmod4,
deep3}
area4=abest
flanth4=tfbest
flanwi4=fbest
webth4=twbest
webdep4=wbest
modval4=smbest
8


8
scanlim=1?
yes
n=maxn
output:
secmod4
modval4
flanth4
flanwi4
webth4
webdep4
go to endlps:
deck:
no
no
lonspac≤2 1/3?
yes
dkpt=0.0772*lonspac+0.17
dkplt=0.12*lonspac+0.07
dkplt≥1.4375
yes
output:
dkplt
return
stand3:
dkplt=
ceil(dkplt/0.0625)*0.0625
onward3:
l≤200?
yes
h3=4
no
200< l <400?
yes
h3=l/50
no
l≥400?
yes
h3=8
no
9

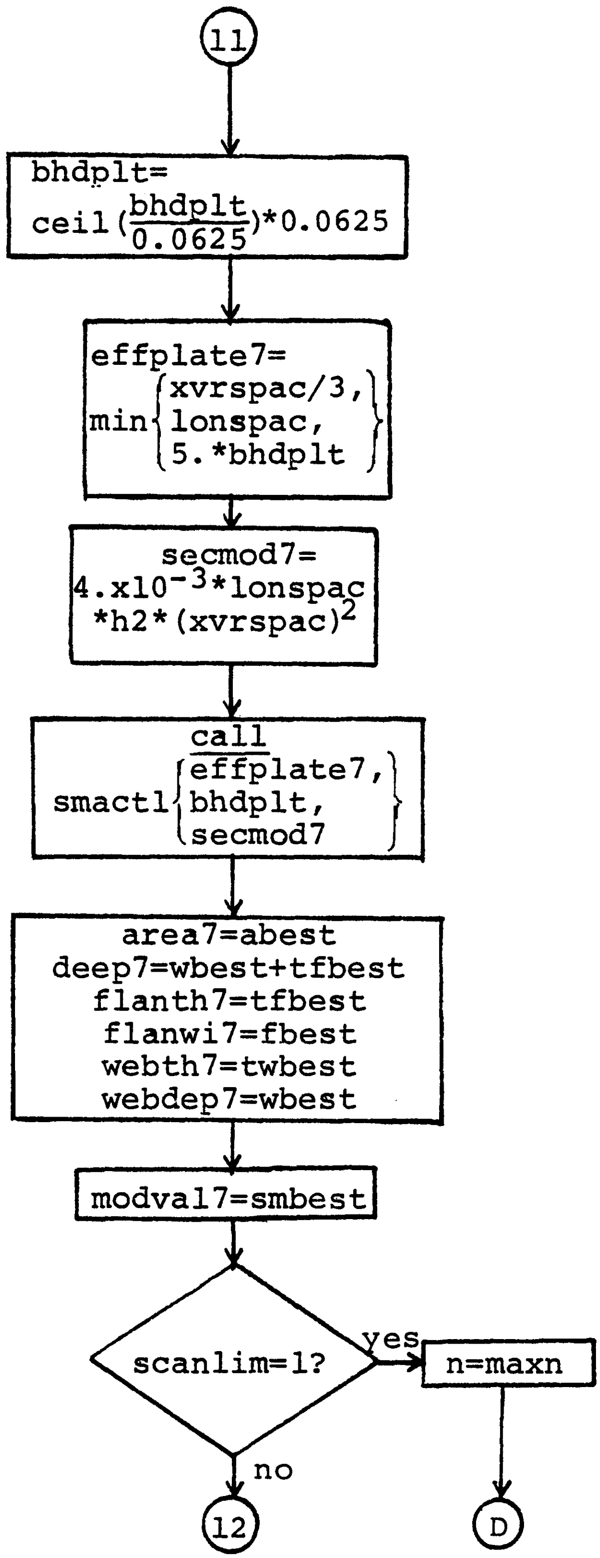
hgmod3:

girder3:

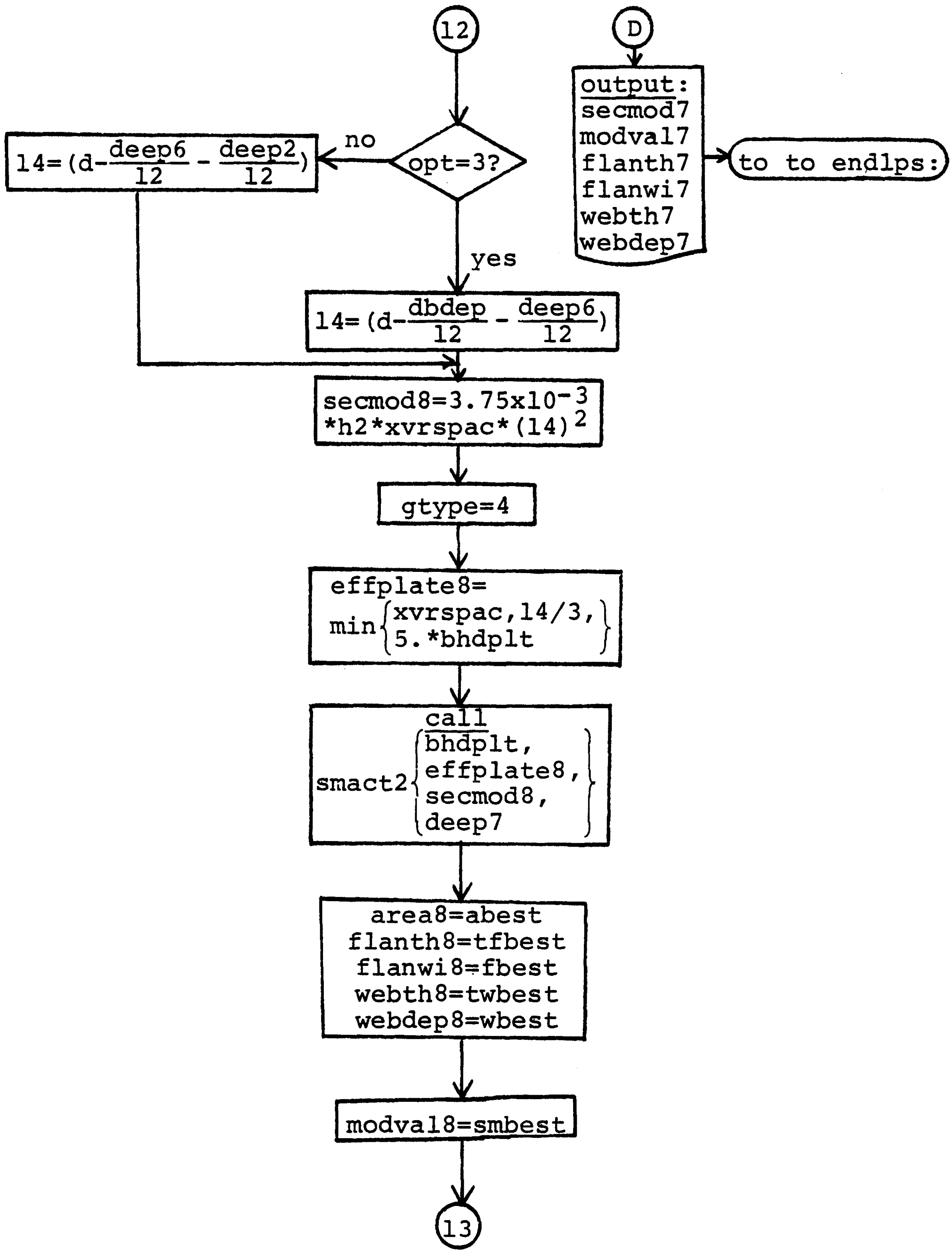


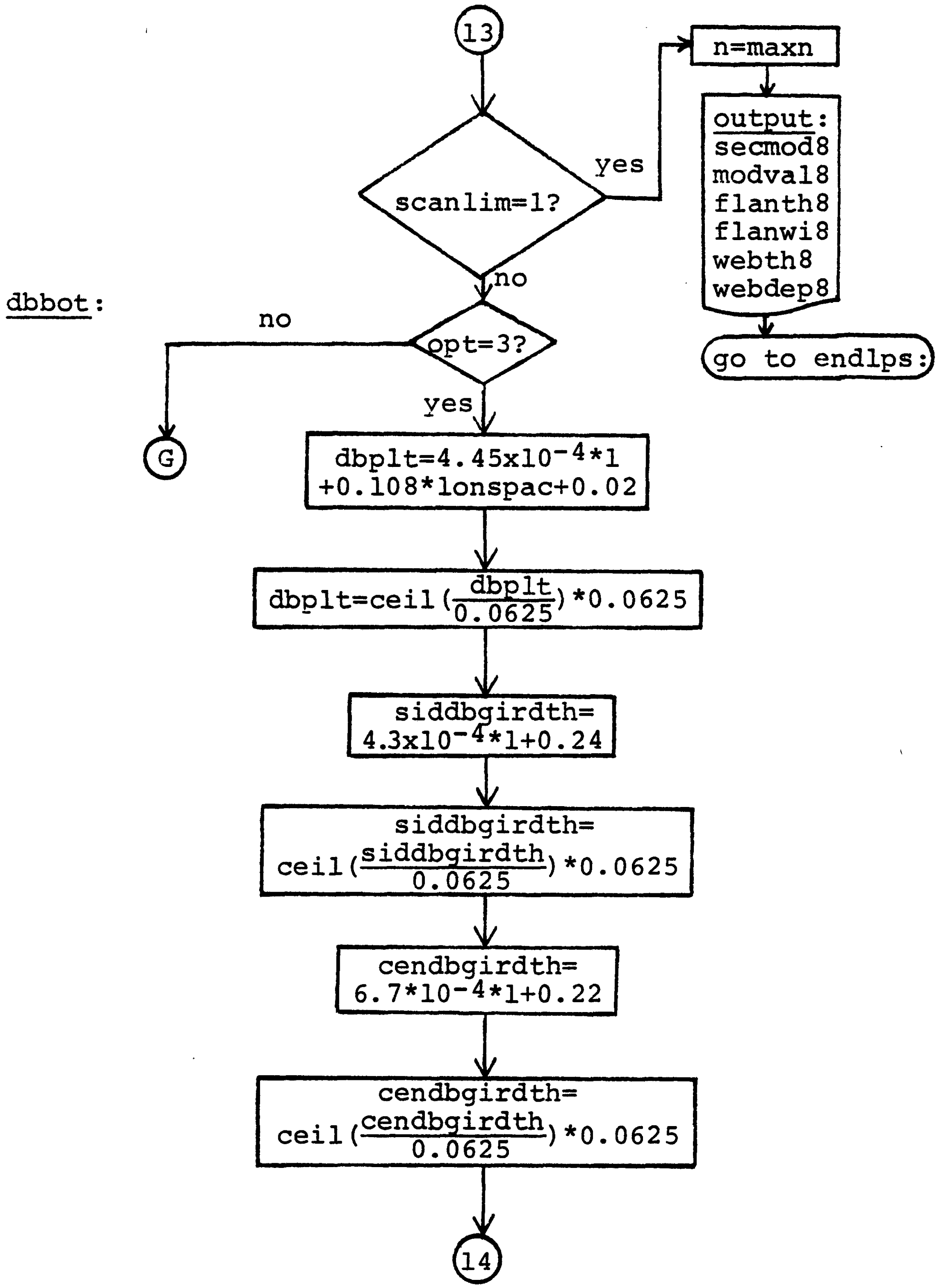

10
effplate6=
min{xvrspac,12/3,
5*dkplt}
gtype=3
call
smact2{dkplt,
effplate6,
secmod6,
deep5}
area6=abest
deep6=wbest+tfbest
flanth6=tfbest
flanwi6=fbest
webth6=twbest
webdep6=wbest
modval6=smbest
scanlim=1?
yes
n=maxn
output:
secmod6
modval6
flanth6
flanwi6
webth6
webdep6
go to
endlps:
no
ihc=1?
yes
go to hulcalc:
no
bhds:
bhdplt=
$\frac{12*\text{lonspac}*\sqrt{d+4-\frac{\text{dbdep}}{12}}+0.1}{460}$
11

stand4:

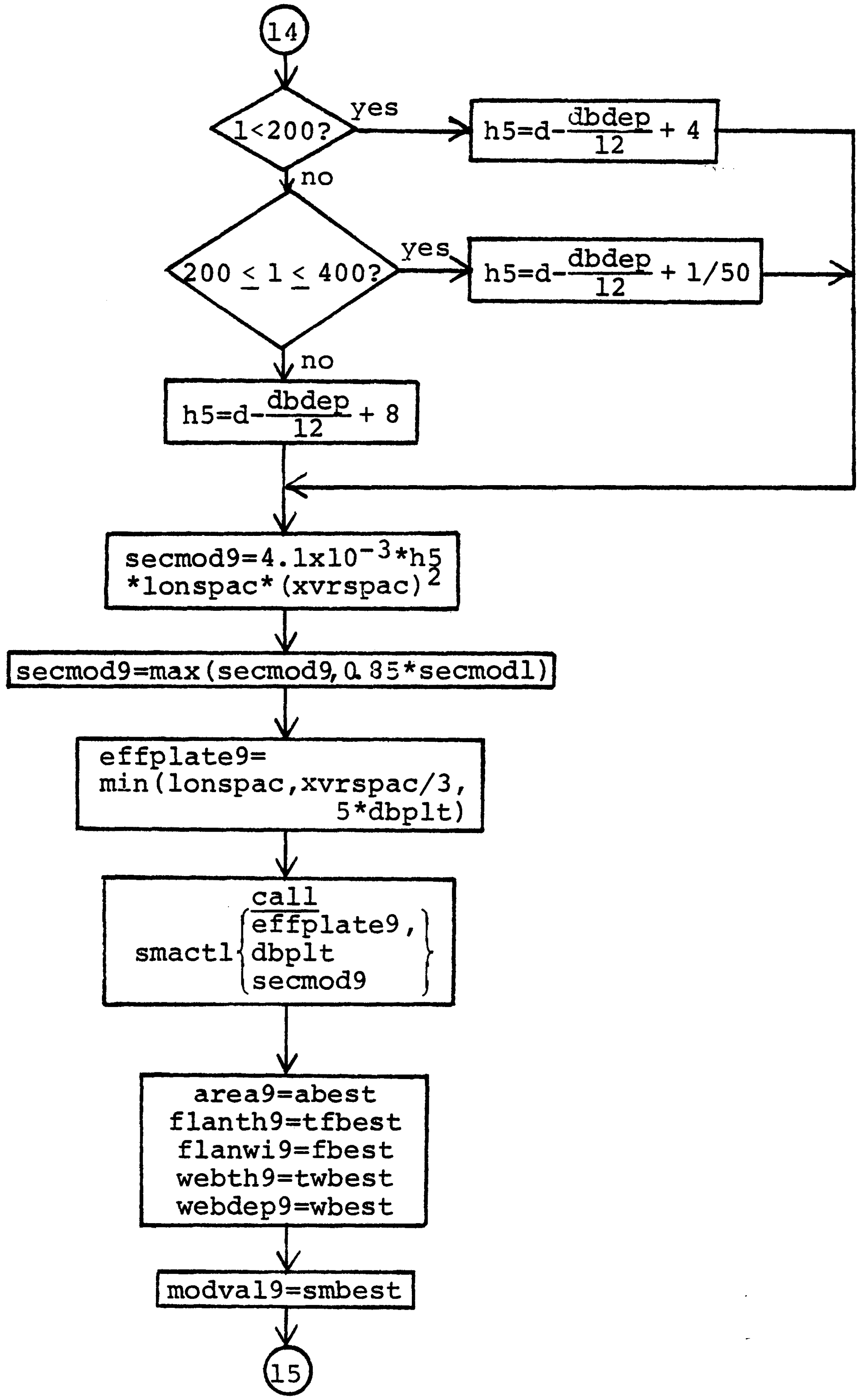


11
bhdplt=
ceil(bhdplt/0.0625)*0.0625
effplate7=
min{xvrspac/3, lonspac, 5.*bhdplt}
secmod7=
4.x10^-3*lonspac
h2(xvrspac)^2
call
smactl{effplate7, bhdplt, secmod7}
area7=abest
deep7=wbest+tfbest
flanth7=tfbest
flanwi7=fbest
webth7=twbest
webdep7=wbest
modval7=smbest
scanlim=1?
yes
n=maxn
no
12
D


12
D
output:
secmod7
modval7
flanth7
flanwi7
webth7
webdep7
to to endlps:
opt=3?
no
yes
14=(d-deep6/12 - deep2/12)
14=(d-dbdep/12 - deep6/12)
secmod8=3.75x10^-3 *h2*xvrspac*(14)^2
gtype=4
effplate8= min{xvrspac,14/3, 5.*bhdplt}
call smact2{bhdplt, effplate8, secmod8, deep7}
area8=abest
flanth8=tfbest
flanwi8=fbest
webth8=twbest
webdep8=wbest
modval8=smbest
13

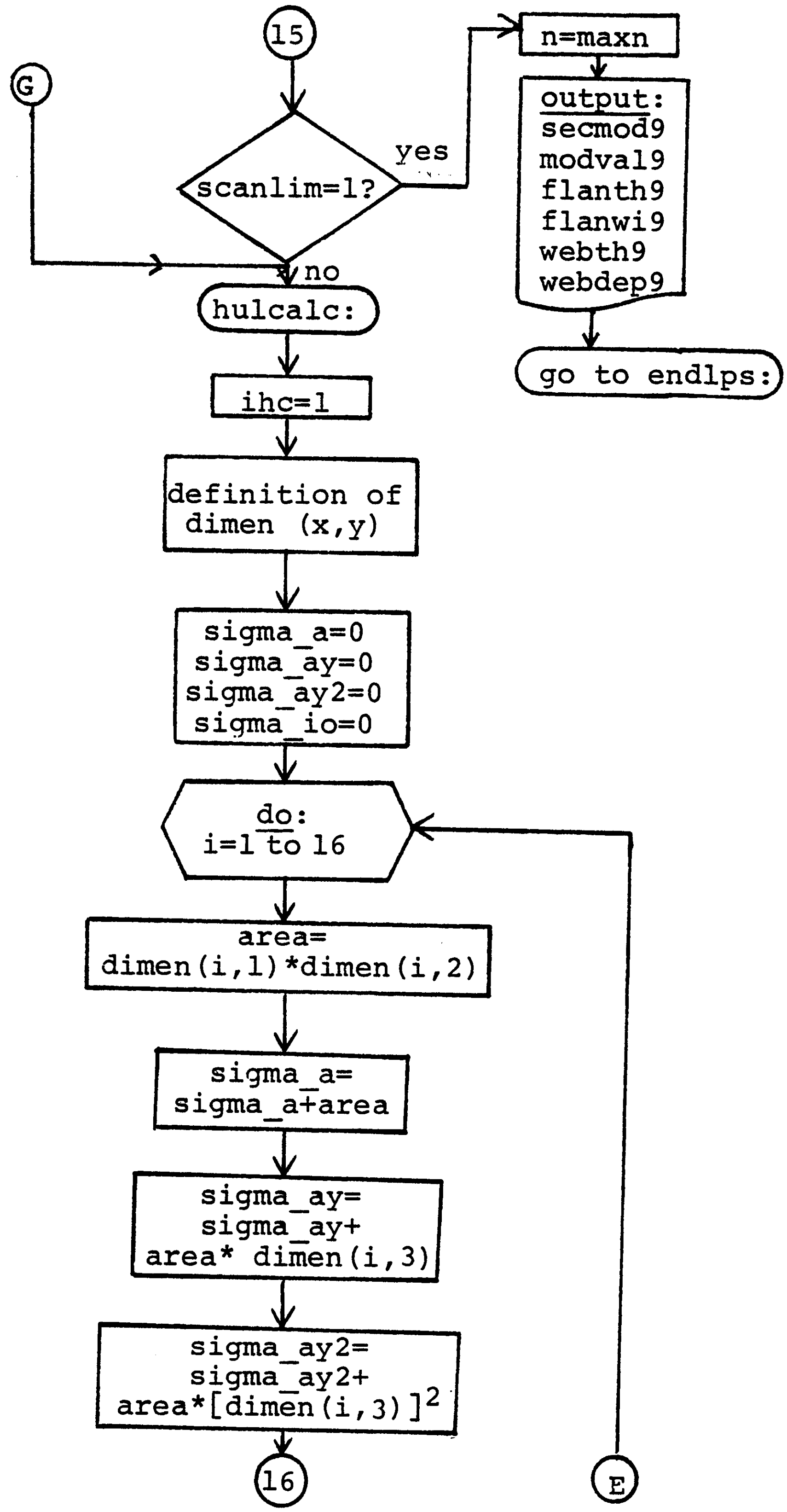


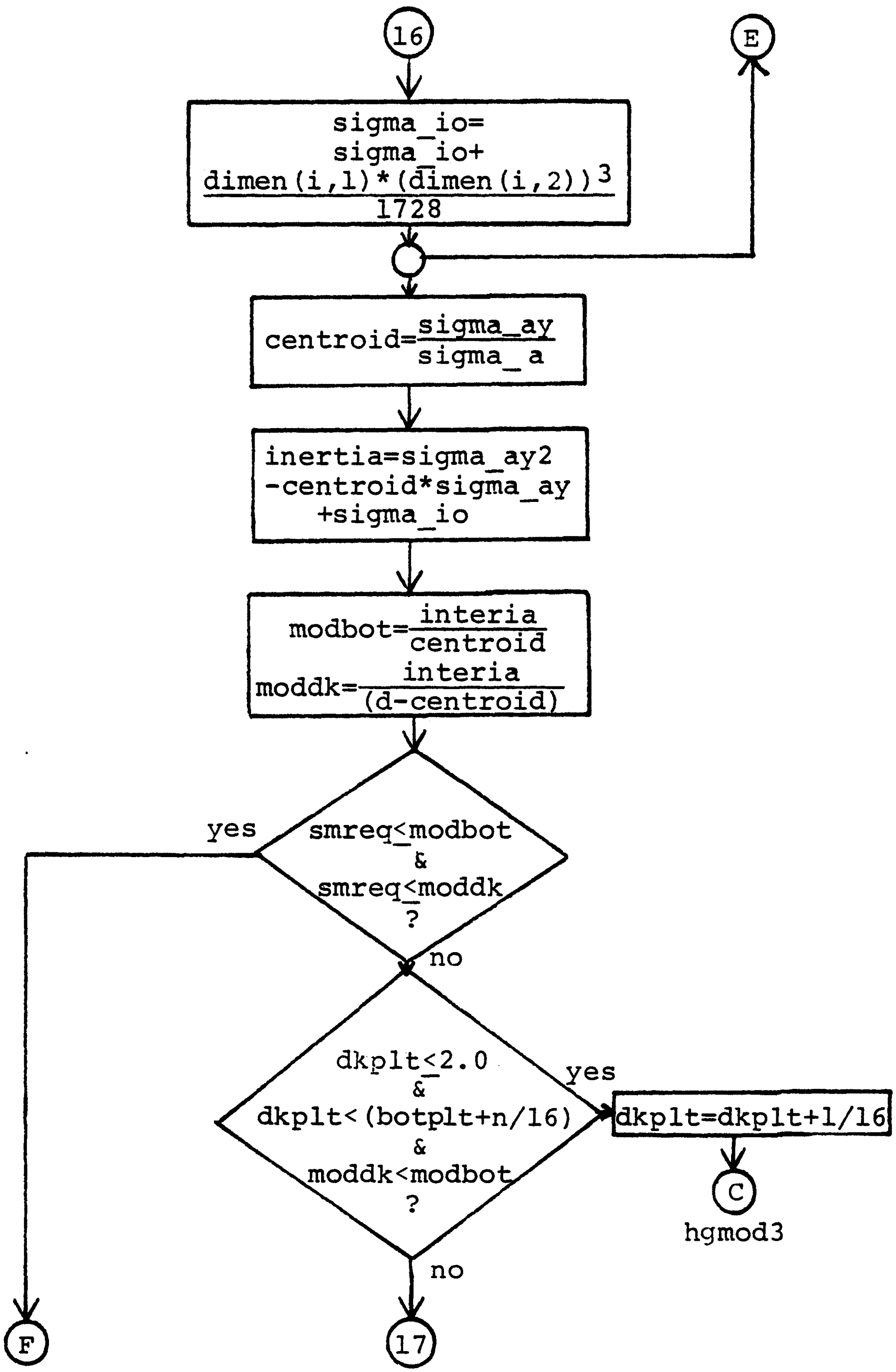
13
scanlim=1?
yes
n=maxn
output:
secmod8
modval8
flanth8
flanwi8
webth8
webdep8
go to endlps:
no
dbbot:
no
opt=3?
G
yes
dbplt=4.45x10^-4*1 +0.108*lonspac+0.02
dbplt=ceil(dbplt/0.0625)*0.0625
siddbgirdth= 4.3x10^-4*1+0.24
siddbgirdth= ceil(siddbgirdth/0.0625)*0.0625
cendbgirdth= 6.7*10^-4*1+0.22
cendbgirdth= ceil(cendbgirdth/0.0625)*0.0625
14

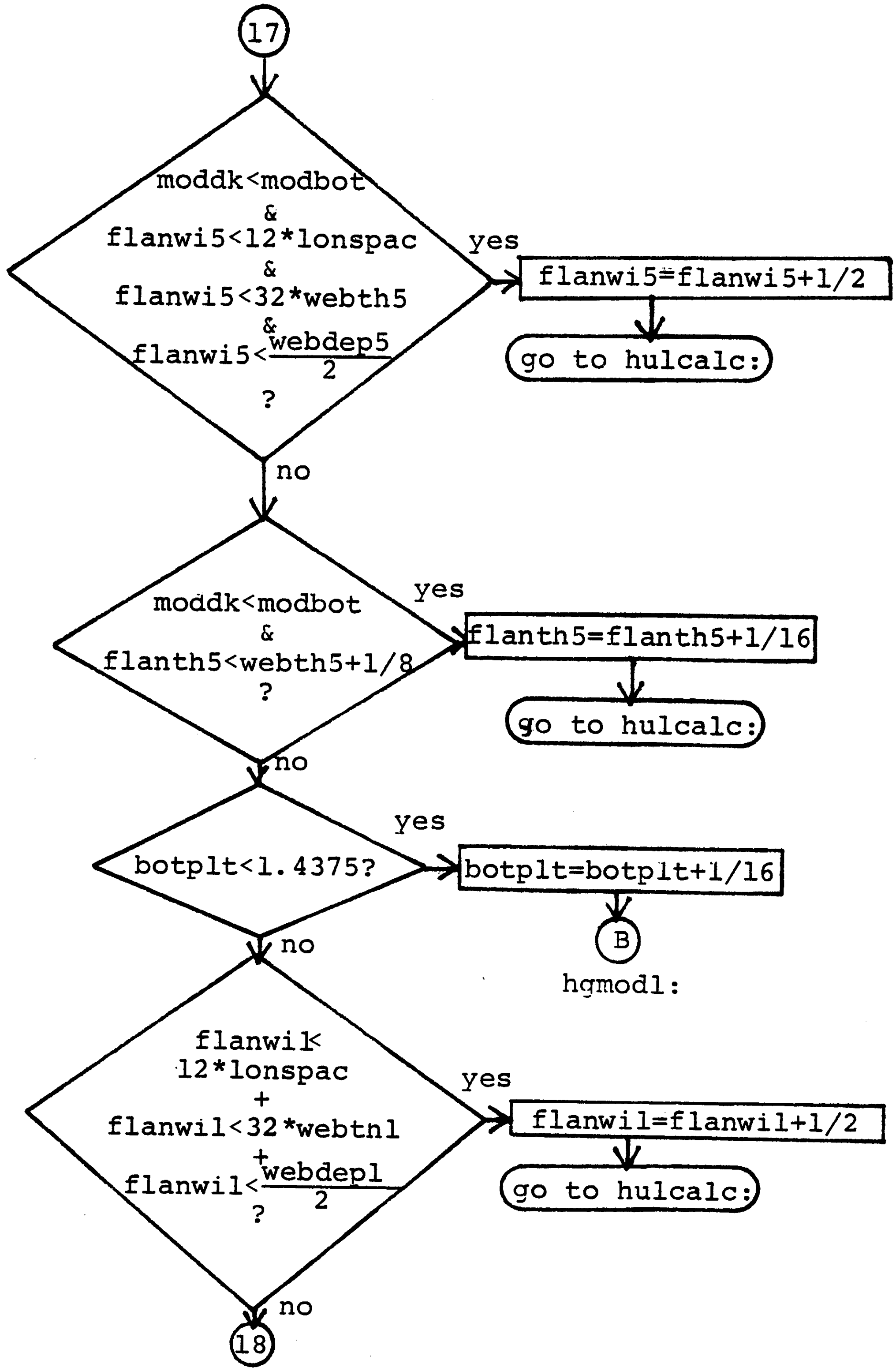

14
l<200?
yes
h5=d-dbdep/12 + 4
no
200 ≤ l ≤ 400?
yes
h5=d-dbdep/12 + 1/50
no
h5=d-dbdep/12 + 8
secmod9=4.1x10^-3*h5
lonspac(xvrspac)^2
secmod9=max(secmod9,0.85*secmodl)
effplate9=
min(lonspac,xvrspac/3,
5*dbplt)
call
smactl
effplate9,
dbplt
secmod9
area9=abest
flanth9=tfbest
flanwi9=fbest
webth9=twbest
webdep9=wbest
modval9=smbest
15

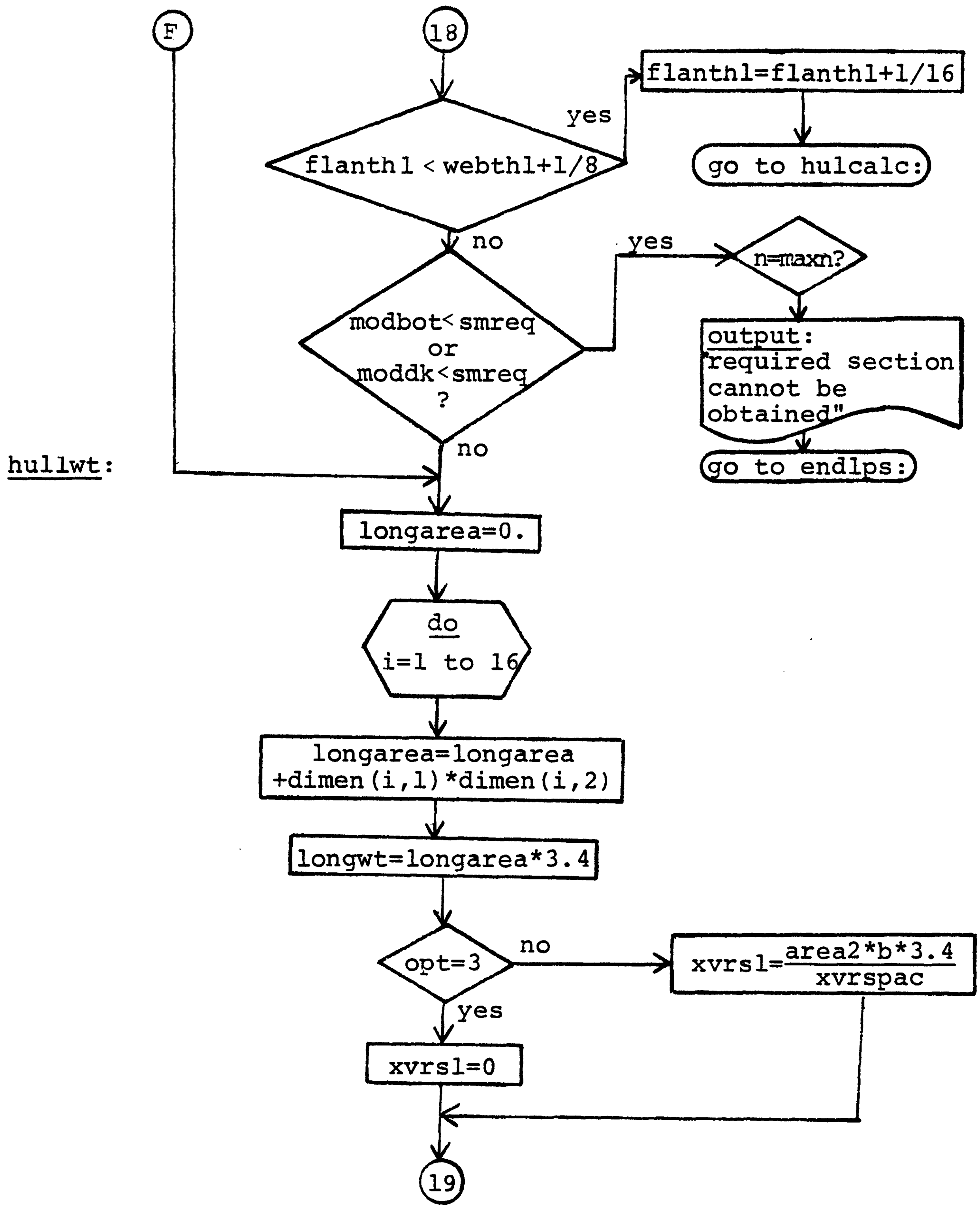
hulcalc:

hullsm:

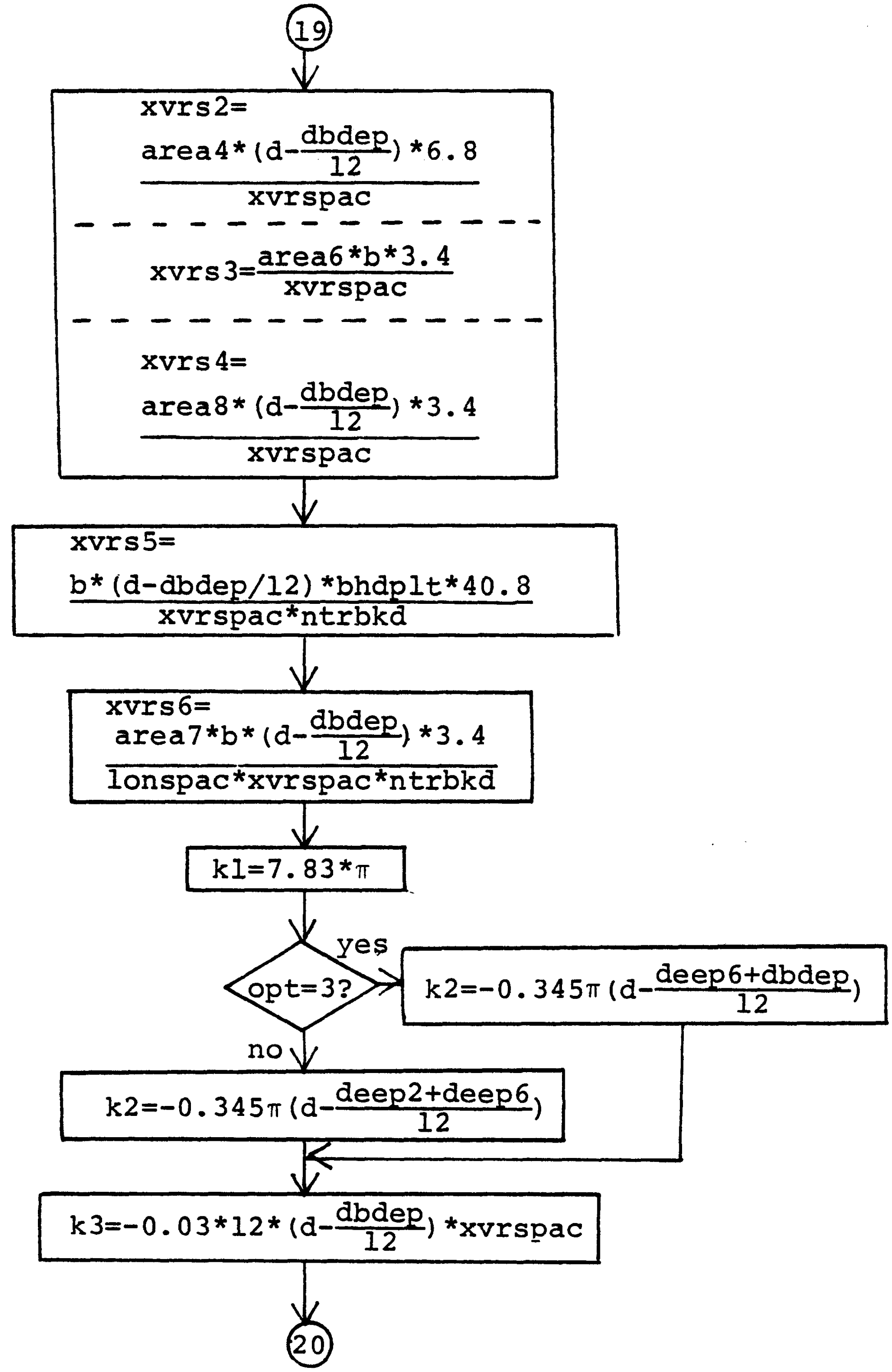


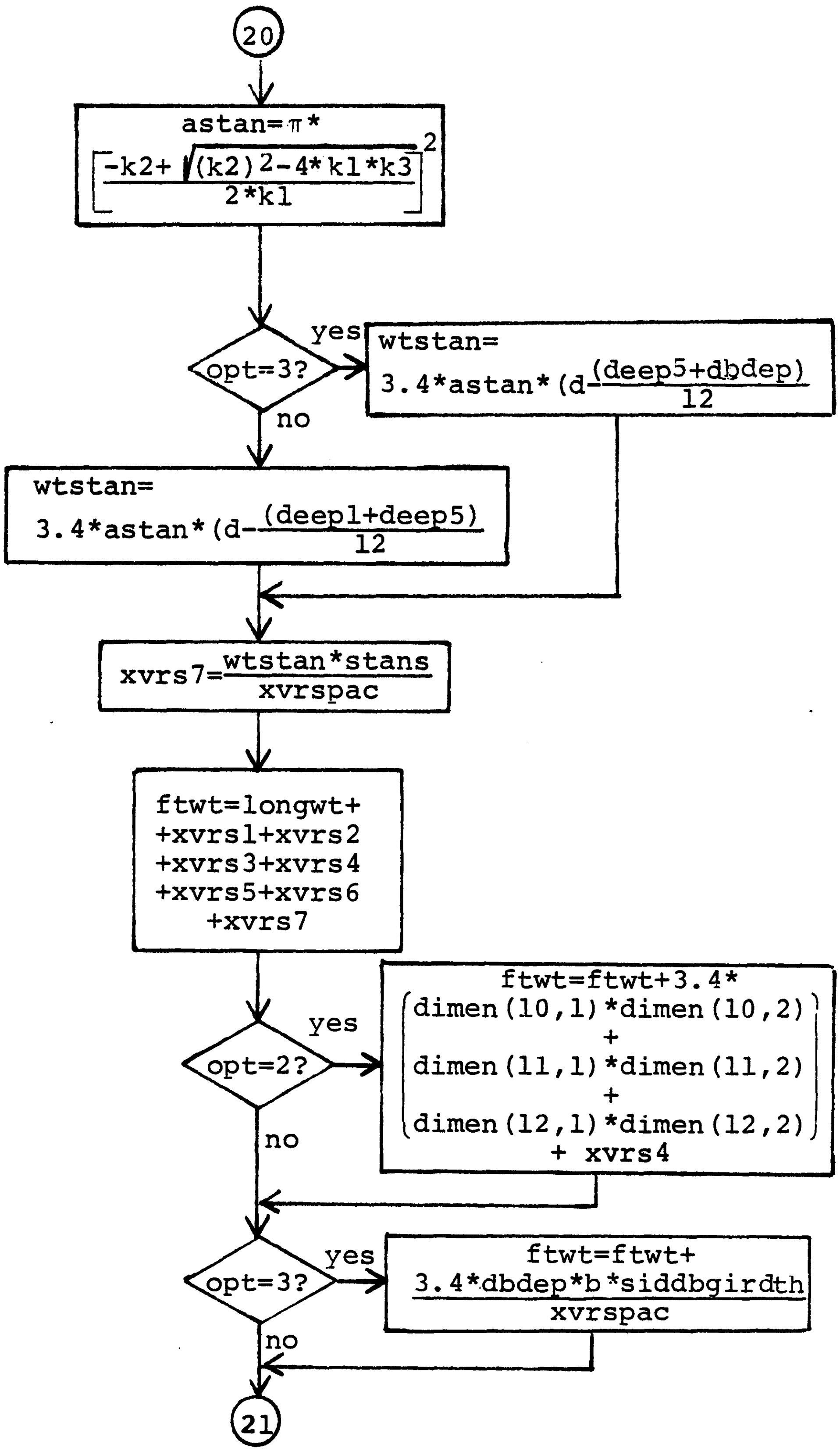

16
E
sigma_io=
sigma_io+
dimen(i,1)*(dimen(i,2))3
1728
centroid=sigma_ay
sigma_a
inertia=sigma_ay2
-centroid*sigma_ay
+sigma_io
modbot=interia
centroid
moddk=interia
(d-centroid)
yes
smreq≤modbot
&
smreq≤moddk
?
no
dkplt≤2.0
&
dkplt<(botplt+n/16)
&
moddk<modbot
?
yes
dkplt=dkplt+1/16
C
hgmod3
no
F
17

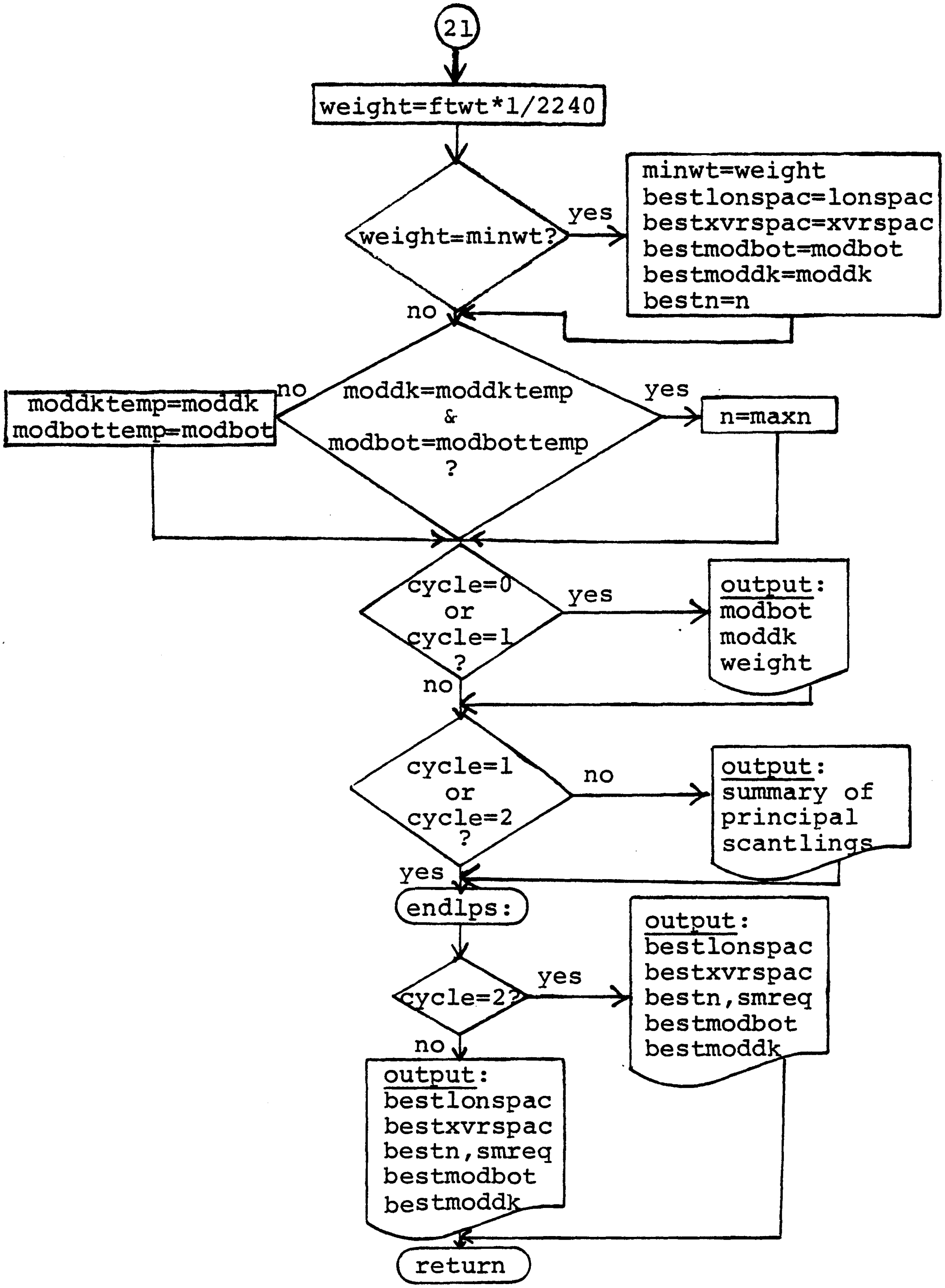

17
moddk<modbot
&
flanwi5<12*lonspac
&
flanwi5<32*webth5
&
flanwi5<webdep5/2
?
yes
flanwi5=flanwi5+1/2
go to hulcalc:
no
moddk<modbot
&
flanth5<webth5+1/8
?
yes
flanth5=flanth5+1/16
go to hulcalc:
no
botplt<1.4375?
yes
botplt=botplt+1/16
B
hgmod1:
no
flanwi1<
12*lonspac
+
flanwi1<32*webtn1
+
flanwi1<webdep1/2
?
yes
flanwi1=flanwi1+1/2
go to hulcalc:
no
18



F
18
flanthl < webthl+1/8
yes
flanthl=flanthl+1/16
go to hulcalc:
no
modbot< smreq
or
moddk<smreq
?
yes
n=maxn?
output:
"required section
cannot be
obtained"
go to endlps:
no
hullwt:
longarea=0.
do
i=1 to 16
longarea=longarea
+dimen(i,1)*dimen(i,2)
longwt=longarea*3.4
opt=3
no
xvrsl=area2*b*3.4/xvrspac
yes
xvrsl=0
19


19
xvrs2= area4*(d-dbdep/12)*6.8 / xvrspac
xvrs3= area6*b*3.4 / xvrspac
xvrs4= area8*(d-dbdep/12)*3.4 / xvrspac
xvrs5= b*(d-dbdep/12)*bhdplt*40.8 / xvrspac*ntrbkd
xvrs6= area7*b*(d-dbdep/12)*3.4 / lonspac*xvrspac*ntrbkd
k1=7.83*π
opt=3?
yes
k2=-0.345π(d-(deep6+dbdep)/12)
no
k2=-0.345π(d-(deep2+deep6)/12)
k3=-0.03*12*(d-dbdep/12)*xvrspac
20


20
astan=π*
[-k2+ √(k2)2-4*k1*k3 / 2*k1]2
opt=3?
yes
no
wtstan=
3.4*astan*(d-(deep5+dbdep) / 12
wtstan=
3.4*astan*(d-(deep1+deep5) / 12
xvrs7=wtstan*stans / xvrspac
ftwt=longwt+
+xvrs1+xvrs2
+xvrs3+xvrs4
+xvrs5+xvrs6
+xvrs7
opt=2?
yes
no
ftwt=ftwt+3.4*
dimen(10,1)*dimen(10,2)
+
dimen(11,1)*dimen(11,2)
+
dimen(12,1)*dimen(12,2)
+ xvrs4
opt=3?
yes
no
ftwt=ftwt+
3.4*dbdep*b*siddbgirdth / xvrspac
21


21
weight=ftwt*1/2240
weight=minwt?
yes
no
minwt=weight
bestlonspac=lonspac
bestxvrspac=xvrspac
bestmodbot=modbot
bestmoddk=moddk
bestn=n
moddk=moddktemp
&
modbot=modbottemp
?
no
yes
moddktemp=moddk
modbottemp=modbot
n=maxn
cycle=0
or
cycle=1
?
yes
no
output:
modbot
moddk
weight
cycle=1
or
cycle=2
?
no
yes
output:
summary of
principal
scantlings
endlps:
cycle=2?
yes
no
output:
bestlonspac
bestxvrspac
bestn,smreq
bestmodbot
bestmoddk
output:
bestlonspac
bestxvrspac
bestn,smreq
bestmodbot
bestmoddk
return

```
bargdes6:  proc(l,b,d,t,cb,minlonspac,maxlonspac,incrlonspac,minxvrspac,maxxvrspac,incrxvrspac,minn,maxn,
                coeff,cycle,opt,minwt,bestlonspac,bestxvrspac,bestn);
           dcl (l,b,d,t,cb,lonspac,xvrspac) dec float;
           dcl (minlonspac, maxlonspac,incrlonspac,minxvrspac,
                maxxvrspac,incrxvrspac) dec float;
           dcl (botplt,h1,secmod1,area1,deep1,flanwi1,flanth1,webth1,
               webdep1,modval1,l2,secmod2,area2,deep2,modval2) dec float;
           dcl (sidplt,h2,secmod3,area3,deep3,flanth3,flanwi3,webth3,
               flanwi2,flanth2,webth2,webdep2,flanth4,flanwi4,webth4,
               webdep4,flanth6,flanwi6,webth6,webdep6,flanth8,flanwi8,
               webth8,webdep8,
               webdep3,modval3,l4,area4,secmod4,modval4) dec float;
           dcl (dkplt,h3,secmod5,area5,deep5,flanth5,flanwi5,webth5,
               webdep5,modval5,secmod6,area6,deep6,modval6) dec float;
           dcl (bhdplt,secmod7,area7,deep7,flanth7,flanwi7,webth7,
               webdep7,modval7,secmod8,area8,modval8) dec float;
           dcl (ltemp,smreq,fact,sigma_a,sigma_ay,sigma_ay2,sigma_io,
               centroid,inertia,modbot,moddk) dec float;
           dcl (longarea,longwt,xvrs1,xvrs2,xvrs3,xvrs4,xvrs5,xvrs6,xvrs7,effplate1,effplate2,effplate3,
               effplate4,effplate5,effplate6,effplate7,effplate8,ftwt,weight,astan,wtstan) dec float;
           dcl (dbplt,dbdep,siddbgirdth,cendbgirdth,h5,effplate9,secmod9,modval9,
                area9,flanth9,flanwi9,webth9,webdep9) dec float;
           dcl (minwt,bestlonspac,bestxvrspac,bestmodbot,bestmoddk) dec float;
           dcl (coeff(66), dimen(16,3)) dec float;
           dcl (abest,area,fbest,smbest,tfbest,twbest,wbest) dec float;
           dcl (scanlim,countr,i,n,pi,ihc,stans,gtype,cycle,opt,minn,maxn,bestopt,bestn,ntrbkd) bin fixed;
          dcl (k1,k2,k3,moddktemp,modbottemp) dec float;
           dcl sysprint file stream output;
           pi=3.141592654;
           minwt=99999.;
interpol: if l=750. then fact=coeff(66);
           else do;
                 countr=ceil((l-99.9999)/10.);
                   fact=coeff(countr)+(l-100-10*(countr-1))*
                        (coeff(countr+1)-coeff(countr))/10;
                end;
onward: smreq=fact*b*(cb+5e-1);
if cycle=0 | cycle=1 then    put skip edit("length=",l,"  beam=",b,"  depth=",d,  "   draft=",
               t,"  required modulus=",smreq) (col(2),a,f(3),a,f(6,2),
                a,f(5,2),a,f(5,2),a,f(6));
return:
if opt^=2 then do;
           if b<=35 then do;
                l2=(b-5.)/2.;
                stans=0;
           end;
           if b>35 & b<=67 then do;
                l2=(b-7.)/4.;
                stans=2;
           end;
           if b>67 & b<=99 then do;
                l2=(b-9.)/6.;
                stans=4;
           end;
           if b>99. then do;
                l2=(b-11.)/8.;
                stans=6;
           end;
           go to loops;
```

```
     end;
     if opt=1 & stans>=2 then do;
          stans=stans-2;
          12=(b-(5.+stans))/(stans+2.);
           go to loops;
     end;
     if opt = 2 then do;
          stans=0;
            12=(b-9.)/3.;
      end;
loops: do lonspac = minlonspac to maxlonspac by incrlonspac;
     do xvrspac = minxvrspac to maxxvrspac by incrxvrspac;
     ntrbkd = floor(0.15*1/xvrspac);
     moddktemp = 0.;
     modbottemp = 0.;
       do n = minn to maxn;
          ihc=0;
          if cycle=0 | cycle =1 then put skip(2) edit("longitudinal frame spacing = ",lonspac,
               "  transverse web spacing = ",xvrspac)(col(2),a,f(5,2),a,f(5,2));
 bottom: if l<400 then
            botplt=5.55e-4*l+8.4e-2*lonspac+6.7e-2;
         if l>=400 then
            botplt=1e-3*l+1.2e-1*lonspac-2e-1;
            if botplt >= 1.4375e0 then
               do;
                  put data(botplt);
                  return;
               end;
stand1:  botplt=ceil(botplt/.0625)*0.0625;
 onward1: if l<=200 then
             h1=d+4;
          if l>200&l<400 then
             h1=d+l/50;
          if l>=400 then
             h1=d+8;
     if opt^=3 then  secmod1=5.494e-3*lonspac*h1*xvrspac*xvrspac;
             else  secmod1=5.33e-3*lonspac*h1*xvrspac*xvrspac;
hgmod1:    effplate1=min(5.*botplt,lonspac,xvrspac/3.);
          call smact1(effplate1,botplt,secmod1);
          area1=abest;
          deep1=wbest+tfbest;
          flanth1=tfbest;
          flanwi1=fbest;
          webth1=twbest;
          webdep1=wbest;
          modval1=smbest;
          if scanlim=1 then do;
               n = maxn;
                put data(secmod1,modval1,flanth1,flanwi1,webth1,
                    webdep1);
               go to endlps;
          end;
girder1:  if opt = 3 then do;
          dbdep = ceil(0.384*b + 4.13*sqrt(t));
          go to side;
          end;
           else dbdep = 0.0;
          secmod2=4.375e-3*h1*xvrspac*12*12;
          gtype=1;
          effplate2=min(xvrspac,12/3.,5.*botplt);
```

```
             call smact2(botplt,effplate2,secmod2,deep1);
             area2=abest;
             deep2=wbest+tfbest;
             flanth2=tfbest;
             flanwi2=fbest;
             webth2=twbest;
             webdep2=wbest;
             modval2=smbest;
             if scanlim=1 then do;
                  n = maxn;
                  put data(secmod2,modval2,flanth2,flanwi2,webth2,webdep2);
                  go to endlps;
             end;
             if ihc = 1 then go to hulcalc;
      side:  if l<=500 then
                sidplt=8.2e-4*l+9e-2*lonspac;
             if l>500 then
                sidplt=3.5e-4*l+1.2e-1*lonspac+1.52e-1;
                if sidplt >= 1.4375e0 then
                   do;
                      put data(sidplt);
                      return;
                   end;
 stand2:  sidplt=ceil(sidplt/.0625)*0.0625;
  onward2: if l<=200. then
                h2=(d-dbdep/12.)/2. + 4.;
             if l>200&l<400 then
                h2=(d-dbdep/12.)/2. + l/50;
             if l>=400 then
                h2=(d-dbdep/12.)/2. + 8.;
             effplate3=min(lonspac,xvrspac/3.,5.*sidplt);
             secmod3=5.125e-3*lonspac*h2*xvrspac*xvrspac;
             call smact1(effplate3,sidplt,secmod3);
             area3=abest;
             deep3=wbest+tfbest;
             flanth3=tfbest;
             flanwi3=fbest;
             webth3=twbest;
             webdep3=wbest;
             modval3=smbest;
             if scanlim=1 then do;
                  n = maxn;
                   put data(secmod3,modval3,flanth3,flanwi3,webth3,webdep3);
                  go to endlps;
             end;
 girder2: if opt =3 then l4 = d -3. - dbdep/12.;
                      else l4 = d-(deep2/12. + 3.);
             effplate4=min(xvrspac,l4/3.,5.*sidplt);
             secmod4=4.375e-3*h2*xvrspac*l4*l4;
             gtype=2;
             call smact2(sidplt,effplate4,secmod4,deep3);
             area4=abest;
             flanth4=tfbest;
             flanwi4=fbest;
             webth4=twbest;
             webdep4=wbest;
             modval4=smbest;
             if scanlim=1 then do;
                  n = maxn;
                  put data(secmod4,modval4,flanth4,flanwi4,webth4,webdep4);
```

```
                go to endlps;
           end;
     deck: if lonspac<=2.3333 then
              dkplt=1.2e-1*lonspac+7e-2;
                        else
              dkplt= 7.72e-2*lonspac+1.7e-1;
              if dkplt >= 1.4375 then
                 do;
                    put data(dkplt);
                    return;
                 end;
stand3:  dkplt=ceil(dkplt/.0625)*0.0625;
 onward3: if l<=200 then
              h3=4;
           if l>200&l<400 then
              h3=l/50;
           if l>=400 then
              h3=8;
           secmod5=5.125e-3*lonspac*h3*xvrspac*xvrspac;
hgmod3: effplate5=min(5.*dkplt,lonspac,xvrspac/3.);
          call smact1(effplate5,dkplt,secmod5);
           area5=abest;
           deep5=wbest+tfbest;
           flanth5=tfbest;
           flanwi5=fbest;
           webth5=twbest;
           webdep5=wbest;
           modval5=smbest;
           if scanlim=1 then do;
                n = maxn;
                 put data(secmod5,modval5,flanth5,flanwi5,webth5,webdep5);
                go to endlps;
           end;
 girder3: secmod6=5e-3*h3*xvrspac*12*12;
           effplate6 = min(xvrspac,12/3.0,5.0*dkplt);
           gtype=3;
           call smact2(dkplt,effplate6,secmod6,deep5);
           area6=abest;
           deep6=wbest+tfbest;
           flanth6=tfbest;
           flanwi6=fbest;
           webth6=twbest;
           webdep6=wbest;
           modval6=smbest;
           if scanlim=1 then do;
                n = maxn;
                put data(secmod6,modval6,flanth6,flanwi6,webth6,webdep6);
                go to endlps;
           end;
           if ihc = 1 then go to hulcalc;
     bhds: bhdplt=12.*lonspac*sqrt(d+4-dbdep/12.)/460.+1.e-1;
stand4:    bhdplt=ceil(bhdplt/.0625)*0.0625;
           effplate7=min(lonspac,xvrspac/3.,5.*bhdplt);
           secmod7=4.1e-3*lonspac*h2*xvrspac*xvrspac;
           call smact1(effplate7,bhdplt,secmod7);
           area7=abest;
           deep7=wbest+tfbest;
           flanth7=tfbest;
           flanwi7=fbest;
           webth7=twbest;
```

```
            webdep7=wbest;
            modval7=smbest;
            if scanlim=1 then do;
                 n = maxn;
                  put data(secmod7,modval7,flanth7,flanwi7,webth7,webdep7);
                 go to endlps;
            end;
girder4:    if opt =3 then l4 = d - dbdep/12. - deep6/12.;
                      else l4 = (d - deep6/12. - deep2/12.);
         secmod8=3.75e-3*h2*xvrspac*l4*l4;
            gtype=4;
            effplate8=min(xvrspac,l4/3.,5.*bhdplt);
            call smact2(bhdplt,effplate8,secmod8,deep7);
            area8=abest;
            flanth8=tfbest;
            flanwi8=fbest;
            webth8=twbest;
            webdep8=wbest;
            modval8=smbest;
            if scanlim=1 then do;
                 n = maxn;
                 put data(secmod8,modval8,flanth8,flanwi8,webth8,webdep8);
                 go to endlps;
            end;
dbbot:  if opt =3 then do;
            dbplt = 4.45e-4*l + 0.108*lonspac + 0.02;
            dbplt = ceil(dbplt/0.0625)*0.0625;
            siddbgirdth = 4.3e-4*l + 0.24;
            siddbgirdth = ceil(siddbgirdth/0.0625)*0.0625;
            cendbgirdth = 6.7e-4*l + 0.22;
            cendbgirdth = ceil(cendbgirdth/0.0625)*0.0625;
            if l<200. then h5 = d - dbdep/12. + 4.;
            if l>=200. & l<=400. then h5 = d - dbdep/12. + 1/50.;
            if l>400. then h5 = d - dbdep/12. + 8.;
            secmod9 = 4.1e-3*h5*lonspac*xvrspac**2;
            secmod9 = max(0.85*secmod1,secmod9);
            effplate9 = min(lonspac,xvrspac/3.,5.*dbplt);
            call smact1(effplate9,dbplt,secmod9);
            area9 = abest;
            flanth9 = tfbest;
            flanwi9 = fbest;
            webth9 = twbest;
            webdep9 = wbest;
            modval9 = smbest;
            if scanlim = 1 then do;
                 put data(secmod9,modval9,flanth9,flanwi9,webth9,webdep9);
                 n = maxn;
                   go to endlps;
            end;
       end;
hulcalc: ihc=1;
            dimen = 0.0;
            dimen(1,1)=12*b;
            dimen(1,2)=botplt;
            dimen(1,3)=botplt/24;
            dimen(2,1)=webth1*b/lonspac;
            dimen(2,2)=webdep1;
            dimen(2,3)=(botplt+webdep1/2)/12;
            dimen(3,1)=flanwi1*b/lonspac;
            dimen(3,2)=flanth1;
```

```
          dimen(3,3)=(botplt+webdep1+flanth1/2)/12;
          dimen(4,1)=2*sidplt;
          dimen(4,2)=12*d;
          dimen(4,3)=d/2;
          dimen(5,1)=webdep3*2*d/lonspac;
          dimen(5,2)=webth3;
          dimen(5,3)=d/2;
          dimen(6,1)=flanth3*2*d/lonspac;
          dimen(6,2)=flanwi3;
          dimen(6,3)=d/2;
          dimen(7,1)=12*b;
          dimen(7,2)=dkplt;
          dimen(7,3)=d-(dkplt/24);
          dimen(8,1)=webth5*b/lonspac;
          dimen(8,2)=webdep5;
          dimen(8,3)=d-(dkplt+webdep5/2)/12;
          dimen(9,1)=flanwi5*b/lonspac;
          dimen(9,2)=flanth5;
          dimen(9,3)=d-(dkplt+webdep5+flanth5/2)/12;
          dimen(10,1)=bhdplt;
          dimen(10,2)=12*d - dbdep;
          dimen(10,3)=d/2. + dbdep/24.;
          dimen(11,1)=webdep7*(d-dbdep/12.)/lonspac;
          dimen(11,2)=webth7;
          dimen(11,3)=d/2 + dbdep/24.;
          dimen(12,1)=flanth7*d/lonspac;
          dimen(12,2)=flanwi7;
          dimen(12,3)=d/2 + dbdep/24.;
     if opt =3 then do;
          dimen(13,1) = 12.*b;
          dimen(13,2) = dbplt;
          dimen(13,3) = (botplt + dbdep + dbplt/2.)/12.;
          dimen(14,1) = webth9*b/lonspac;
          dimen(14,2) = webdep9;
          dimen(14,3) = (botplt + dbdep - webdep9/2.)/12.;
          dimen(15,1) = flanwi9*b/lonspac;
          dimen(15,2) = flanth9;
          dimen(15,3) = (botplt + dbdep -webdep9 - flanth9/2.)/12.;
          dimen(16,1) = 2*siddbgirdth*(ceil(b/30.) - 1) + cendbgirdth;
          dimen(16,2) = dbdep;
          dimen(16,3) = (botplt + dbdep/2.)/12.;
     end;
hullsm: sigma_a=0;
        sigma_ay=0;
        sigma_ay2=0;
        sigma_io=0;
        do i=1 to 16;
           area=dimen(i,1)*dimen(i,2);
           sigma_a=sigma_a+area;
           sigma_ay=sigma_ay+area*dimen(i,3);
           sigma_ay2=sigma_ay2+area*dimen(i,3)*dimen(i,3);
           sigma_io=sigma_io+dimen(i,1)*dimen(i,2)**3/1728;
        end;
        centroid=sigma_ay/sigma_a;
        inertia=sigma_ay2+sigma_io-centroid*sigma_ay;
        modbot=inertia/centroid;
        moddk=inertia/(d-centroid);
        if smreq<=modbot&smreq<=moddk then
           go to hullwt;
         if dkplt<(botplt + n*0.0625) & moddk<modbot & dkplt<2. then do;
```

```
                 dkplt=dkplt+.0625;
                 go to hgmod3;
              end;
           if flanwi5<webdep5/2 & flanwi5<32.*webth5 & moddk<modbot & flanwi5<lonspac*12. then
              do;
                 flanwi5=flanwi5+.5;
                 go to hulcalc;
              end;
           if flanth5<(webth5+0.125)&moddk<modbot then
              do;
                 flanth5=flanth5+.0625;
                 go to hulcalc;
              end;
           if botplt<1.4375  then
              do;
                 botplt=botplt+.0625;
                 go to hgmod1;
              end;
           if flanwi1<webdep1/2 & flanwi1<32.*webth1  & flanwi1<12.*lonspac then
              do;
                 flanwi1=flanwi1+.5;
                 go to hulcalc;
              end;
           if flanth1<(webth1+0.125)  then
              do;
                 flanth1=flanth1+.0625;
                 go to hulcalc;
              end;
           if modbot<smreq|moddk<smreq then do;
              if n= maxn then put edit("required section modulus cannot be obtained",
                 " with reasonable scantlings increases",lonspac,xvrspac)(col(2),a,a,2 f(6,2));
            go to endlps;
            end;
           put skip(2);
           put edit("results of hull section modulus calculations")(col(15),a);
            go to endlps;
 hullwt: longarea=0;
           do i=1 to 16;
              longarea=longarea+dimen(i,1)*dimen(i,2);
           end;
           longwt=longarea*3.4;
if opt^=3 then  xvrs1=area2*b*3.4/xvrspac;
            else xvrs1 = 0.;
           xvrs2=area4*(d - dbdep/12.)*6.8/xvrspac;
           xvrs3=area6*b*3.4/xvrspac;
           xvrs4=area8*(d - dbdep/12.)*3.4/xvrspac;
           xvrs5=b*(d-dbdep/12.)*bhdplt*40.8/(xvrspac*ntrbkd);
           xvrs6=area7*(b/lonspac)*(d-dbdep/12.)*3.4/(xvrspac*ntrbkd);
            k1=7.83*pi;
if opt =3 then k2 = -0.345*pi*(d-(deep6+dbdep)/12.);
            else k2 = -0.345*pi*(d-(deep2+deep6)/12.);
            k3=-0.03*12*(d-dbdep/12.)*xvrspac;
            astan=pi*((-k2+sqrt(k2*k2-4*k1*k3))/(2*k1))**2;
if opt ^= 3 then wtstan = 3.4*astan*(d-(deep1+deep5)/12.);
              else wtstan = 3.4*astan*(d-(deep5+dbdep)/12.);
            xvrs7=wtstan*stans/xvrspac;
           ftwt=longwt+xvrs1+xvrs2+xvrs3+xvrs4+xvrs5+xvrs6+xvrs7;
            if opt = 2 then
               ftwt=ftwt+3.4*(dimen(10,1)*dimen(10,2)+dimen(11,1)*dimen(11,2)+dimen(12,1)*dimen(12,2))-.vrs4;
            if opt = 3 then ftwt = ftwt + 3.4*dbdep*siddbgirdth*b/xvrspac;
```

```
        weight=ftwt*1/2240.;
     if weight<minwt then do;
          minwt=weight;
          bestlonspac=lonspac;
          bestxvrspac=xvrspac;
          bestmodbot=modbot;
          bestmoddk=moddk;
          bestn=n;
     end;
     if moddk=moddktemp & modbot=modbottemp then n=maxn;
       else do;
           modbottemp=modbot;
           moddktemp=moddk;
      end;
          if cycle=0 | cycle=1 then put skip edit("bottom modulus=",modbot,"deck modulus=",moddk,
                  "estimated bare hull weight=",weight)
                 (col(2),3 (a,f(6),x(2)));
smact1: procedure(s,tp,smr);
        dcl (tp,tw,w,tf,f,p,s,smr,ap,aw,af,yp,yw,yf,ym,im,smf,am) dec float;
        if tp>=.375 then
           tw=tp-.125;
           else tw=tp;
        w=3.5;
        tf=tp;
        f = max(tw,0.750);
        wbest,twbest,fbest,tfbest,smf,smbest=0;
        scanlim=0.0;
        p=12*s;
smcalc: ap=p*tp;
        aw=w*tw;
        af=f*tf;
        yp=tp/2;
        yw=tp+w/2;
        yf=tp+w+tf/2;
        ym=(ap*yp+aw*yw+af*yf)/(ap+aw+af);
        im=ap*yp*yp+aw*yw*yw+af*yf*yf+(tw*w*w*w)/12-
           ym*(ap*yp+aw*yw+af*yf);
        smf=im/(tp+w+tf-ym);
        if smf>=smr then
 loop1:    do;
              am=aw+af;
                    abest=am;
                    wbest=w;
                    twbest=tw;
                    fbest=f;
                    tfbest=tf;
                    smbest=smf;
                    go to endsm1;
           end;
        if w<18.&w<5*f & w<75.*tw then
           do;
              w=w+.5;
              go to smcalc;
           end;
        if f=tw then f=ceil(f/0.25)*0.25;
        if f<w/2 &f<32.*tw & f<lonspac*12. then
           do;
              f=f+.5;
              go to smcalc;
           end;
```

```
         if tf<tp then
            do;
               tf=tf+.0625;
               go to smcalc;
            end;
         if tw<tp then
            do;
               tw=tw+.0625;
               go to smcalc;
            end;
         else do;
                 put edit("scantling limits reached in smact1")(col(2),a);
                 if cycle = 2 then put data(lonspac,xvrspac,n);
                 scanlim=1;
                 go to loop1;
              end;
endsm1: end smact1;
 smact2: procedure(tp,s,smr,dep);
          dcl (tp,tw,w,tf,f,s,smr,ap,aw,af,yp,yw,yf,ym,im,smf,am,smp,dep) dec float;
          scanlim=0.0;
         if gtype=2 | gtype=4 then w=1.5*14;
         if gtype=1 then w=2.4*12;
         if gtype=3 then w=1.5*12;
         w=ceil(max(w,2.5*dep)/.25)*0.25;
         if l<=200 then tw=0.34;
         if l>200&l<270 then tw=0.34+(l-200)/3500;
         if l>=270&l<700 then tw=0.36 + (l-270)/3000;
         if l>=700 then tw=0.50;
         if gtype=4 then tw=min(0.44,1.2e-1+1.e-2*w);
         if tp >= 0.625 then tw = max(tw, tp - .25);
                     else tw = max(tw, tp);
         tw=ceil(max(tw,w/75)/.0625)*0.0625;
          tf=tp;
         f = ceil(min(w/5.0,20.0*tw)/0.25)*0.25;
          wbest,twbest,fbest,tfbest,smf,smp,smbest=0;
 smcalc: ap=12*s*tp;
          aw=w*tw;
          af=f*tf;
          yp=tp/2;
          yw=tp+w/2;
          yf=tp+w+tf/2;
          ym=(ap*yp+aw*yw+af*yf)/(ap+aw+af);
          im=ap*yp*yp+aw*yw*yw+af*yf*yf+(tw*w*w*w)/12-
             ym*(ap*yp+aw*yw+af*yf);
          smf=im/(tp+w+tf-ym);
          smp=im/ym;
          if smf>=smr & smp>=smr then
loop1:        do;
                am=aw+af;
                      abest=am;
                      wbest=w;
                      twbest=tw;
                      fbest=f;
                      tfbest=tf;
                      if smf>=smp then
                         smbest=smp;
                      else smbest=smf;
         if ((gtype=1 | gtype =3) & w>48.) | w>88. then do;
              put skip list("transverse web debth limit exceeded for girder type:",gtype);
              scanlim = 1.0;
```

```
                if cycle=2 then n=maxn;
                 return;
          end;
                        go to endsm2;
                end;
if w<47. & w<=5.*f & w<=75.*tw then
                do;
                   w=w+1;
                   go to smcalc;
                end;
             if f<=w/2.0 & f<=32.*tw then
                do;
                   f=f+1;
                   go to smcalc;
                end;
              if tf<(tw+0.5) & tf<1.4375 then do;
                     tf = tf + 0.0625;
                      go to smcalc;
               end;
              if tw<(tp+0.5) & tw<1.4375 then do
                     tw = tw + 0.0625;
                     go to smcalc;
              end;
if ((gtype=2 | gtype=4) & w<87.) & w<=5.*f & w<=75.*tw then do;
        w = w + 1;
        go to smcalc;
end;
            if gtype=4 &  tw<1.4375 then do;
                tw = tw + 0.0625;
                go to smcalc;
            end;
             else do;
                    put edit("scantlings limits reached in smact2")(col(2),a);
                    if cycle = 2 then put data(lonspac,xvrspac,n);
                    scanlim=1;
                    go to loop1;
                  end;
  endsm2: end smact2;
     if cycle = 1 | cycle = 2 then go to endlps;
          put skip(2) edit("summary of principal scantlings")(col(20),a);
          put skip(2);
           put edit("bottom shell - ",botplt," in. side shell - ",sidplt,
                " in.  bulkhead plating - ",bhdplt," in.  deck plate - ",dkplt)
                (col(2),a,f(6,4),a,f(6,4),a,f(6,4),a,f(6,4));
if opt =3 then put edit(" in. doub-bot shell - ",dbplt) (a,f(6,4),a);
          put edit("bottom longitudinals: flange- ",flanwi1," x ",flanth1,
               ";   web- ",webdep1," x ",webth1)(col(2),a,x(2),f(5,2),a,f(6,4),
               a,f(5,2),a,f(6,4));
if opt^=3 then put edit("bottom transverses: flange- ",flanwi2," x ",flanth2,
               ";   web- ",webdep2," x ",webth2)(col(2),a,x(4),f(5,2),a,f(6,4),
               a,f(5,2),a,f(6,4));
               put edit("side longitudinals: flange- ",flanwi3," x ",flanth3,
                   ";   web- ",webdep3," x ",webth3)(col(2),a,x(4),f(5,2),a,f(6,4),
                   a,f(5,2),a,f(6,4));
          put edit("side transverses: flange- ",flanwi4," x ",flanth4,
               ";   web- ",webdep4," x ",webth4)(col(2),a,x(6),f(5,2),a,f(6,4),
               a,f(5,2),a,f(6,4));
              put edit("bulkhead longitudinals: flange- ",flanwi7," x ",flanth7,
                  ";   web- ",webdep7," x ",webth7)(col(2),a,f(5,2),a,f(6,4),
                  a,f(5,2),a,f(6,4));
```

```
          put edit("bulkhead transverses: flange- ",flanwi8," x ",flanth8,
              ";   web- ",webdep8," x ",webth8)(col(2),a,x(2),f(5,2),a,f(6,4),
              a,f(5,2),a,f(6,4));
                 put edit("deck longitudinals: flange- ",flanwi5," x ",flanth5,
                     ";   web- ",webdep5," x ",webth5)(col(2),a,x(4),f(5,2),a,f(6,4),
                     a,f(5,2),a,f(6,4));
          put edit("deck transverses: flange- ",flanwi6," x ",flanth6,
              ";   web- ",webdep6," x ",webth6)(col(2),a,x(6),f(5,2),a,f(6,4),
              a,f(5,2),a,f(6,4));
if opt = 3 then put edit("doub-bott longitudinals: flange-",flanwi9," x ",flanth9,";   web- ",webdep9,
                     " x ",webth9) (col(2),a,f(5,2),a,f(6,4),a,f(5,2),a,f(6,4));
 endlps: end; end; end;
if cycle=2 then put edit(bestlonspac,bestxvrspac,bestn,smreq,bestmodbot,bestmoddk)(f(5,2),f(4),x(3),f(2),3 f(6));
           else put data(bestlonspac,bestxvrspac,bestn,smreq,bestmodbot,bestmoddk);
        return;
end bargdes6;
```

NAMES DECLARED IN THIS COMPILATION.

IDENTIFIER	OFFSET	LOC	STORAGE CLASS	DATA TYPE	ATTRIBUTES AND REFERENCES
NAMES DECLARED BY DECLARE STATEMENT.					
abest		001070	automatic	float dec(10)	dcl 25 set ref 93 121 153 171 201 217 235 253 278 461 529
af		000122	automatic	float dec(10)	dcl 498 set ref 517 521 521 522 522 524
af		000125	automatic	float dec(10)	dcl 438 set ref 450 454 454 455 455 463
am		000152	automatic	float dec(10)	dcl 438 set ref 460 461
am		000147	automatic	float dec(10)	dcl 498 set ref 528 529
ap		000117	automatic	float dec(10)	dcl 438 set ref 448 454 454 455 455
ap		000114	automatic	float dec(10)	dcl 498 set ref 515 521 521 522 522
area		001073	automatic	float dec(10)	dcl 25 set ref 347 348 349 350
area1		000117	automatic	float dec(10)	dcl 6 set ref 99
area2		000152	automatic	float dec(10)	dcl 6 set ref 121 401
area3		000174	automatic	float dec(10)	dcl 8 set ref 153
area4		000304	automatic	float dec(10)	dcl 8 set ref 171 403
area5		000326	automatic	float dec(10)	dcl 13 set ref 201
area6		000356	automatic	float dec(10)	dcl 13 set ref 217 404
area7		000375	automatic	float dec(10)	dcl 15 set ref 235 407
area8		000425	automatic	float dec(10)	dcl 15 set ref 253 405
area9		000623	automatic	float dec(10)	dcl 21 set ref 278
astan		000565	automatic	float dec(10)	dcl 19 set ref 412 413 414
aw		000117	automatic	float dec(10)	dcl 498 set ref 516 521 521 522 522 528
aw		000122	automatic	float dec(10)	dcl 438 set ref 449 454 454 455 455 462
b			parameter	float dec(10)	dcl 3 ref 1 37 38 43 44 47 47 43 51 51 52 55 56 63 68 113 292 295 298 310 313 316 329 332 335 338 401 404 405 407 419
bestlonspac			parameter	float dec(10)	dcl 23 set ref 1 423 612 613
bestmodcot		000642	automatic	float dec(10)	dcl 23 set ref 425 612 613
bestmodk		000645	automatic	float dec(10)	dcl 23 set ref 426 612 613
bestn		001126	automatic	fixed bin(17,0)	dcl 26 set ref 427 612 613
bestxvrspac			parameter	float dec(10)	dcl 23 set ref 1 424 612 613
bndplt		000367	automatic	float dec(10)	dcl 15 set ref 230 231 231 232 234 251 252 319 406 581
botplt		000106	automatic	float dec(10)	dcl 6 set ref 79 81 83 85 88 88 97 98 119 120 293 294 297 300 331 334 337 340 359 373 375 375 581
cb			parameter	float dec(10)	dcl 3 ref 1 37
cenbgirdth		000604	automatic	float dec(10)	dcl 21 set ref 269 270 270 339
centroid		000460	automatic	float dec(10)	dcl 17 set ref 353 354 355 356
coeff			parameter	float dec(10)	array dcl 24 ref 1 31 34 34 34
countr		001116	automatic	fixed bin(17,0)	dcl 26 set ref 33 34 34 34 34
cycle			parameter	fixed bin(17,0)	dcl 26 ref 1 38 38 77 77 434 434 578 578 612 492 540 573
d			parameter	float dec(10)	dcl 3 ref 1 38 89 91 93 144 146 149 165 166 230 247 248 271 272 273 302 303 304 306 307 309 312 315 318 320 321 322 324 325 327 356 403 405 406 407 409 410 411 413 414
dbdep		000576	automatic	float dec(10)	dcl 21 set ref 113 116 144 146 148 165 230 247 271 272 273 320 321 322 324 327 331 334 337 339 340 403 405 406 407 409 411 414 419
dbplt		000573	automatic	float dec(10)	dcl 21 set ref 265 266 266 276 277 330 331 584
deep1		000122	automatic	float dec(10)	dcl 6 set ref 100 120 413
deep2		000155	automatic	float dec(10)	dcl 6 set ref 122 166 248 410
deep3		000177	automatic	float dec(10)	dcl 8 set ref 154 170
deep5		000331	automatic	float dec(10)	dcl 13 set ref 202 216 413 414
deep6		000361	automatic	float dec(10)	dcl 13 set ref 218 247 248 409 410
deep7		000403	automatic	float dec(10)	dcl 15 set ref 236 252

```
dep                               parameter    float dec(10)    dcl 498 ref 497 503
dimen                      000050 automatic    float dec(10)    array dcl 24 set ref 291 292 293 294 295 296 297
                                                                298 299 300 301 302 303 304 305 306 307 308 309
                                                                310 311 312 313 314 315 316 317 318 319 320 321
                                                                322 323 324 325 326 327 329 330 331 332 333 334
                                                                335 336 337 338 339 340 347 347 349 350 350 351
                                                                351 398 399 417 417 417 417 417 417
dkplt                      000315 automatic    float dec(10)    dcl 13 set ref 182 184 186 188 191 191 193 203
                                                                214 216 311 312 315 318 359 359 360 360 581
effplate1                  000527 automatic    float dec(10)    dcl 19 set ref 97 98
effplate2                  000532 automatic    float dec(10)    dcl 19 set ref 119 120
effplate3                  000535 automatic    float dec(10)    dcl 19 set ref 150 152
effplate4                  000540 automatic    float dec(10)    dcl 19 set ref 167 170
effplate5                  000543 automatic    float dec(10)    dcl 19 set ref 199 200
effplate6                  000546 automatic    float dec(10)    dcl 19 set ref 214 216
effplate7                  000551 automatic    float dec(10)    dcl 19 set ref 232 234
effplate8                  000554 automatic    float dec(10)    dcl 19 set ref 251 252
effplate9                  000612 automatic    float dec(10)    dcl 21 set ref 276 277
f                          000111 automatic    float dec(10)    dcl 498 set ref 513 517 532 545 550 550 552 552
                                                                563
f                          000111 automatic    float dec(10)    dcl 438 set ref 444 450 464 469 474 474 474 475
                                                                475 475 477 477
fact                       000441 automatic    float dec(10)    dcl 17 set ref 31 34 37
fbest                      001076 automatic    float dec(10)    dcl 25 set ref 102 124 156 173 204 220 238 255
                                                                280 445 464 514 532
flantn1                    000130 automatic    float dec(10)    dcl 6 set ref 101 108 299 300 383 385 385 385
flantn2                    000216 automatic    float dec(10)    dcl 8 set ref 123 130 588
flantn3                    000202 automatic    float dec(10)    dcl 8 set ref 155 162 307 591
flantn4                    000227 automatic    float dec(10)    dcl 8 set ref 172 179 594
flantn5                    000334 automatic    float dec(10)    dcl 13 set ref 203 210 317 318 368 370 370 603
flantn6                    000243 automatic    float dec(10)    dcl 8 set ref 219 226 606
flantn7                    000403 automatic    float dec(10)    dcl 15 set ref 237 244 325 597
flantn8                    000257 automatic    float dec(10)    dcl 8 set ref 254 261 600
flantn9                    000626 automatic    float dec(10)    dcl 21 set ref 279 285 336 337 609
flanwi1                    000125 automatic    float dec(10)    dcl 6 set ref 102 108 298 378 378 378 380 380
                                                                585
flanwi2                    000213 automatic    float dec(10)    dcl 8 set ref 124 130 588
flanwi3                    000205 automatic    float dec(10)    dcl 8 set ref 156 162 308 591
flanwi4                    000232 automatic    float dec(10)    dcl 8 set ref 173 179 594
flanwi5                    000337 automatic    float dec(10)    dcl 13 set ref 204 210 316 363 363 363 365 365
                                                                603
flanwi6                    000246 automatic    float dec(10)    dcl 8 set ref 220 226 606
flanwi7                    000406 automatic    float dec(10)    dcl 15 set ref 238 244 326 597
flanwi8                    000262 automatic    float dec(10)    dcl 8 set ref 255 261 600
flanwi9                    000631 automatic    float dec(10)    dcl 21 set ref 280 285 335 609
ftwt                       000557 automatic    float dec(10)    dcl 19 set ref 416 417 417 419 419 420
gtype                      001124 automatic    fixed bin(17,0)  dcl 26 set ref 118 169 215 250 500 500 501 502
                                                                508 537 537 538 563 563 567
h1                         000111 automatic    float dec(10)    dcl 6 set ref 89 91 93 95 96 117
h2                         000166 automatic    float dec(10)    dcl 8 set ref 144 146 148 151 168 233 249
h3                         000320 automatic    float dec(10)    dcl 13 set ref 192 194 196 198 213
h5                         000607 automatic    float dec(10)    dcl 21 set ref 271 272 273 274
i                          001117 automatic    fixed bin(17,0)  dcl 26 set ref 346 347 347 349 350 350 351 351
                                                                397 398 398
inc                        001122 automatic    fixed bin(17,0)  dcl 26 set ref 76 133 229 290
im                         000144 automatic    float dec(10)    dcl 438 set ref 455 457
im                         000141 automatic    float dec(10)    dcl 498 set ref 522 524 525
incrlonspac                       parameter    float dec(10)    dcl 4 ref 1 70
incrxvrspac                       parameter    float dec(10)    dcl 4 ref 1 71
inertia                    000463 automatic    float dec(10)    dcl 17 set ref 354 355 356
```

k1	001130 automatic	float dec(10)	dcl 27 set ref 408 412 412
k2	001133 automatic	float dec(10)	dcl 27 set ref 409 410 412 412 412
k3	001136 automatic	float dec(10)	dcl 27 set ref 411 412
l	parameter	float dec(10)	dcl 3 ref 1 31 33 34 38 72 79 79 81 81 89 91 91 91 93 134 134 136 136 144 146 146 146 148 192 194 194 194 195 265 267 269 271 272 272 272 273 420 504 505 505 505 506 506 506 507
l2	000144 automatic	float dec(10)	dcl 6 set ref 44 48 52 56 63 68 117 117 119 213 213 214 411 501 502
l4	000331 automatic	float dec(10)	dcl 8 set ref 16 . 166 167 168 168 247 248 249 249 251 500
longarea	000474 automatic	float dec(10)	dcl 19 set ref 396 398 398 400
longwt	000477 automatic	float dec(10)	dcl 19 set ref 400 416
lonspac	000100 automatic	float dec(10)	dcl 3 set ref 70 77 79 81 95 96 97 134 136 150 151 182 182 184 198 199 230 232 233 265 274 276 295 298 304 307 313 316 322 325 332 335 363 378 389 407 423 475 492 573
maxlonspac	parameter	float dec(10)	dcl 4 ref 1 70
maxn	parameter	fixed bin(17,0)	dcl 26 ref 1 75 107 129 161 178 209 225 243 260 286 389 429 540
maxxvrspac	parameter	float dec(10)	dcl 4 ref 1 71
minlonspac	parameter	float dec(10)	dcl 4 ref 1 70
minn	parameter	fixed bin(17,0)	dcl 26 ref 1 75
minwt	parameter	float dec(10)	dcl 23 set ref 1 30 421 422
minxvrspac	parameter	float dec(10)	dcl 4 ref 1 71
modnot	000466 automatic	float dec(10)	dcl 17 set ref 355 357 359 363 368 388 425 429 431 434
modnottemp	001144 automatic	float dec(10)	dcl 27 set ref 74 429 431
modjk	000471 automatic	float dec(10)	dcl 17 set ref 356 357 359 363 368 389 426 429 432 434
modjktemp	001141 automatic	float dec(10)	dcl 27 set ref 73 429 432
modval1	000141 automatic	float dec(10)	dcl 6 set ref 105 108
modval2	000160 automatic	float dec(10)	dcl 6 set ref 127 130
modval3	000276 automatic	float dec(10)	dcl 8 set ref 153 162
modval4	000312 automatic	float dec(10)	dcl 8 set ref 176 179
modval5	000350 automatic	float dec(10)	dcl 13 set ref 207 210
modval6	000364 automatic	float dec(10)	dcl 13 set ref 223 226
modval7	000417 automatic	float dec(10)	dcl 15 set ref 241 244
modval8	000430 automatic	float dec(10)	dcl 15 set ref 258 261
modval9	000620 automatic	float dec(10)	dcl 21 set ref 283 285
n	001120 automatic	fixed bin(17,0)	dcl 26 set ref 75 107 129 161 178 209 225 243 260 286 353 389 427 429 492 540 573
ntrokd	001127 automatic	fixed bin(17,0)	dcl 26 set ref 72 406 407
opt	parameter	fixed bin(17,0)	dcl 26 ref 1 41 61 66 95 112 165 247 264 328 401 409 413 417 419 584 588 609
p	000114 automatic	float dec(10)	dcl 438 set ref 447 448
pi	001121 automatic	fixed bin(17,0)	dcl 26 set ref 29 408 409 410 412
s	parameter	float dec(10)	dcl 498 ref 497 515
s	parameter	float dec(10)	dcl 438 ref 437 447
scanlim	001115 automatic	fixed bin(17,0)	dcl 26 set ref 106 128 160 177 208 224 242 259 284 446 493 499 539 574
secmod1	000114 automatic	float dec(10)	dcl 6 set ref 95 96 98 108 275
secmod2	000147 automatic	float dec(10)	dcl 6 set ref 117 120 130
secmod3	000171 automatic	float dec(10)	dcl 8 set ref 151 152 162
secmod4	000307 automatic	float dec(10)	dcl 8 set ref 168 170 179
secmod5	000323 automatic	float dec(10)	dcl 13 set ref 198 200 210
secmod6	000353 automatic	float dec(10)	dcl 13 set ref 213 216 226
secmod7	000372 automatic	float dec(10)	dcl 15 set ref 233 234 244
secmod8	000422 automatic	float dec(10)	dcl 15 set ref 249 252 261
secmod9	000615 automatic	float dec(10)	dcl 21 set ref 274 275 275 277 285

sigbwidth	000601 automatic	float dec(10)	dcl 21 set ref 267 268 268 338 419
sidbit	000163 automatic	float dec(10)	dcl 8 set ref 134 136 138 140 143 143 150 152 167 170 301 581
sigma_a	000444 automatic	float dec(10)	dcl 17 set ref 342 344 348 353
sigma_av	000447 automatic	float dec(10)	dcl 17 set ref 343 349 349 353 354
sigma_av2	000452 automatic	float dec(10)	dcl 17 set ref 344 350 350 354
sigma_io	000455 automatic	float dec(10)	dcl 17 set ref 345 351 351 354
smbest	001101 automatic	float dec(10)	dcl 25 set ref 105 127 159 176 207 223 241 258 283 445 466 514 534 536
smf	000147 automatic	float dec(10)	dcl 438 set ref 445 457 458 466
smf	000144 automatic	float dec(10)	dcl 498 set ref 514 524 526 534 536
smp	000152 automatic	float dec(10)	dcl 498 set ref 514 525 526 534 534
smr	parameter	float dec(10)	dcl 438 ref 437 458
smr	parameter	float dec(10)	dcl 498 ref 497 526 526
smreq	000436 automatic	float dec(10)	dcl 17 set ref 37 39 357 357 388 389 612 613
stans	001123 automatic	fixed bin(17,0)	dcl 26 set ref 45 49 53 57 61 62 62 63 63 67 415
sysprint	000016 constant	file	output stream dcl 29 set ref 26 38 77 85 138 130 140 162 179 198 210 226 244 261 285 389 393 394 434 579 580 581 584 585 588 591 594 597 603 603 606 609 612 613 491 492 538 572 573
t	parameter	float dec(10)	dcl 3 ref 1 38 113
tf	000106 automatic	float dec(10)	dcl 498 set ref 512 517 520 524 533 555 555 556 556
tf	000106 automatic	float dec(10)	dcl 438 set ref 443 450 453 457 465 480 482 482
tfbest	001104 automatic	float dec(10)	dcl 25 set ref 100 101 122 123 154 155 172 202 203 218 219 236 237 254 279 445 465 514 533
tp	parameter	float dec(10)	dcl 438 ref 437 439 439 441 443 448 451 452 453 457 480 485
tp	parameter	float dec(10)	dcl 498 ref 497 509 509 510 512 515 518 519 520 524 559
tw	000100 automatic	float dec(10)	dcl 498 set ref 504 505 506 507 508 509 509 513 510 511 511 513 516 522 531 545 553 555 559 559 559 559 563 567 568 568
tw	000100 automatic	float dec(10)	dcl 438 set ref 439 441 444 449 455 463 469 474 475 485 487 497
twbest	001107 automatic	float dec(10)	dcl 25 set ref 103 125 157 174 205 221 239 256 281 445 463 514 531
w	000103 automatic	float dec(10)	dcl 438 set ref 442 449 452 453 455 455 455 457 462 469 469 469 471 471 475
w	000103 automatic	float dec(10)	dcl 498 set ref 500 501 502 503 503 504 511 513 516 519 520 522 522 522 524 530 537 537 545 545 545 547 547 550 563 563 563 564 564
wbest	001112 automatic	float dec(10)	dcl 25 set ref 100 104 122 126 154 158 175 202 206 218 222 236 240 257 282 445 462 514 530
webjep1	000136 automatic	float dec(10)	dcl 6 set ref 104 139 296 297 300 378 585
webjep2	000224 automatic	float dec(10)	dcl 8 set ref 126 130 548
webjep3	000273 automatic	float dec(10)	dcl 8 set ref 158 162 314 591
webjep4	000240 automatic	float dec(10)	dcl 3 set ref 175 179 594
webjep5	000345 automatic	float dec(10)	dcl 13 set ref 206 210 314 315 318 363 603
webjep6	000254 automatic	float dec(10)	dcl 8 set ref 222 226 606
webjep7	000414 automatic	float dec(10)	dcl 15 set ref 240 244 322 597
webjep8	000270 automatic	float dec(10)	dcl 8 set ref 257 261 608
webjep9	000637 automatic	float dec(10)	dcl 21 set ref 282 285 333 334 337 609
webth1	000133 automatic	float dec(10)	dcl 6 set ref 103 108 295 378 383 585
webth2	000221 automatic	float dec(10)	dcl 8 set ref 125 130 588
webth3	000210 automatic	float dec(10)	dcl 8 set ref 157 162 305 591
webth4	000235 automatic	float dec(10)	dcl 6 set ref 174 179 594
webth5	000342 automatic	float dec(10)	dcl 13 set ref 205 210 313 365 368 603
webth6	000251 automatic	float dec(10)	dcl 8 set ref 221 226 606
webth7	000411 automatic	float dec(10)	dcl 15 set ref 239 244 323 597

```
weoth8            000265 automatic    float dec(10)     dcl 8 set ref 256 261 600
weoth9            000034 automatic    float dec(10)     dcl 21 set ref 281 285 332 609
weight            000562 automatic    float dec(10)     dcl 19 set ref 420 421 422 434
wtstan            000570 automatic    float dec(10)     dcl 19 set ref 413 414 415
xvrs1             000502 automatic    float dec(10)     dcl 19 set ref 401 402 416
xvrs2             000505 automatic    float dec(10)     dcl 19 set ref 403 416
xvrs3             000510 automatic    float dec(10)     dcl 19 set ref 404 416
xvrs4             000513 automatic    float dec(10)     dcl 19 set ref 405 416 417
xvrs5             000516 automatic    float dec(10)     dcl 19 set ref 406 416
xvrs6             000521 automatic    float dec(10)     dcl 19 set ref 407 416
xvrs7             000524 automatic    float dec(10)     dcl 19 set ref 415 416
xvrspac           000103 automatic    float dec(10)     dcl 3 set ref 71 72 77 95 95 96 96 97 117 119
                                                        150 151 151 167 168 198 198 199 213 214 232 233
                                                        233 249 251 274 276 389 401 403 404 405 406 407
                                                        411 415 419 424 492 573
yf                000133 automatic    float dec(10)     dcl 498 set ref 520 521 522 522 522
yf                000136 automatic    float dec(10)     dcl 438 set ref 453 454 455 455 455
ym                000136 automatic    float dec(10)     dcl 498 set ref 521 522 524 525
ym                000141 automatic    float dec(10)     dcl 438 set ref 454 455 457
yp                000130 automatic    float dec(10)     dcl 438 set ref 451 454 455 455 455
yp                000125 automatic    float dec(10)     dcl 498 set ref 518 521 522 522 522
yw                000133 automatic    float dec(10)     dcl 438 set ref 452 454 455 455 455
yw                000130 automatic    float dec(10)     dcl 498 set ref 519 521 522 522 522

NAMES DECLARED BY DECLARE STATEMENT AND NEVER REFERENCED.
bestopt           001125 automatic    fixed bin(17,0)   dcl 26
ftemp             000433 automatic    float dec(10)     dcl 17

NAMES DECLARED BY EXPLICIT CONTEXT.
bargdeso          001220 constant     entry             external dcl 1 ref 1
bncs              004647 constant     label             dcl 230 ref 230
bottom            002320 constant     label             dcl 79 ref 79
bobjt             005436 constant     label             dcl 264 ref 264
deck              004114 constant     label             dcl 182 ref 182
endips            013120 constant     label             dcl 611 ref 110 131 163 180 211 227 245 262 287
                                                        391 395 578 611
endsm1            014236 constant     label             dcl 496 ref 467 496
endsm2            015072 constant     label             dcl 577 ref 543 577
girder1           002724 constant     label             dcl 112 ref 112
girder2           003651 constant     label             dcl 165 ref 165
girder3           004442 constant     label             dcl 213 ref 213
girder4           005154 constant     label             dcl 247 ref 247
ngmod1            002545 constant     label             dcl 97 ref 97 376
ngmod3            004261 constant     label             dcl 199 ref 199 361
nulcalc           006124 constant     label             dcl 290 ref 133 229 290 366 371 381 386
nullsm            007172 constant     label             dcl 342 ref 342
nullwt            010006 constant     label             dcl 396 ref 357 396
interpol          001261 constant     label             dcl 31 ref 31
locp1             013747 constant     label             dcl 458 ref 458 494
loop1             015227 constant     label             dcl 526 ref 526 575
loops             002036 constant     label             dcl 70 ref 59 64 70
onward            001427 constant     label             dcl 37 ref 37
onward1           002430 constant     label             dcl 89 ref 89
onward2           003511 constant     label             dcl 144 ref 144
onward3           004177 constant     label             dcl 192 ref 192
return            001617 constant     label             dcl 41 ref 41
side              003205 constant     label             dcl 134 ref 114 134
smact1            013332 constant     entry             internal dcl 437 ref 98 152 200 234 277 437
smact2            014240 constant     entry             internal dcl 497 ref 120 170 216 252 497
smcalc            014707 constant     label             dcl 515 ref 515 548 553 557 561 565 569
```

```
smcalc                        013452 constant        label                 dcl 443 ref 443 472 478 483 498
stand1                        002412 constant        label                 dcl 88 ref 88
stand2                        003273 constant        label                 dcl 143 ref 143
stand3                        004161 constant        label                 dcl 191 ref 191
stand4                        004737 constant        label                 dcl 231 ref 231

NAMES DECLARED BY CONTEXT OR IMPLICATION.
ceil                                                 builtin function      internal ref 33 88 113 143 191 231 266 268 270
                                                                           338 474 503 511 513
floor                                                builtin function      internal ref 72
max                                                  builtin function      internal ref 275 444 503 509 510 511
min                                                  builtin function      internal ref 97 119 150 167 199 214 232 251 276
                                                                           508 513
sqrt                                                 builtin function      internal ref 113 230 412

STORAGE REQUIREMENTS FOR THIS PROGRAM.

          Object    Text     Link     Symbol   Defs     Static
Start          0       0    16044     16066   15753     16054
Length     24042   15753       22      5737      71         2

BLOCK NAME                  STACK SIZE     TYPE               WHY NONQUICK/WHO SHARES STACK FRAME
barjdese                           914 external procedure  is an external procedure.
smact1                             298 internal procedure  contains a format statement, and uses I/O statements.
smact2                             301 internal procedure  contains a format statement, and uses I/O statements.

THE FOLLOWING EXTERNAL OPERATORS ARE USED BY THIS PROGRAM.
fx1_to_fl2          r_l_a               r_g_a               r_e_as              r_le_a              r_ge_a
call_ext_out        call_int_this       return              ext_entry           int_entry           put_end
stream_io           put_data_eis        put_list_eis        real_to_real_rd     dsqrt               ceil_dec
floor_dec           put_field           put_field_chk       put_control

THE FOLLOWING EXTERNAL ENTRIES ARE CALLED BY THIS PROGRAM.
decimal_exp_

THE FOLLOWING EXTERNAL VARIABLES ARE USED BY THIS PROGRAM.
sysprint                      sysprint.fsb


   LINE    LOC      LINE    LOC      LINE    LOC      LINE    LOC      LINE    LOC      LINE    LOC      LINE    LOC
      1 001204        28 001240        29 001252        30 001254        31 001261        33 001276        34 001322
     37 001427        38 001446        41 001617        43 001623        44 001630        45 001643        47 001644
     48 001657        49 001672        51 001674        52 001707        53 001722        55 001724        56 001730
     57 001743        59 001745        61 001746        62 001757        63 001761        64 002015        66 002016
     67 002021        68 002022        70 002036        71 002111        72 002165        73 002213        74 002216
     75 002221        76 002230        77 002231        79 002320        81 002346        83 002372        85 002376
     86 002411        88 002412        89 002430        91 002441        93 002470        95 002501        96 002525
     97 002545        98 002606        99 002620       100 002625       101 002631       102 002635       103 002642
    104 002646       105 002652       106 002656       107 002661       108 002664       110 002723       112 002724
    113 002730       114 002774       116 002775       117 003000       118 003020       119 003022       120 003062
    121 003076       122 003102       123 003106       124 003112       125 003117       126 003123       127 003127
    128 003134       129 003137       130 003142       131 003201       133 003202       134 003205       136 003227
    138 003253       140 003257       141 003272       143 003273       144 003311       146 003344       148 003415
    150 003450       151 003511       152 003531       153 003543       154 003547       155 003553       156 003557
    157 003564       158 003571       159 003576       160 003603       161 003606       162 003611       163 003650
    165 003651       166 003676       167 003716       168 003756       169 003776       170 004000       171 004014
    172 004020       173 004025       174 004031       175 004035       176 004041       177 004046       178 004051
    179 004054       180 004113       182 004114       184 004131       186 004141       188 004145       189 004160
```

B.3 DICTIONARY OF VARIABLES USED IN THE BARGE DESIGN MODEL

DEFINITION OF VARIABLES IN THE SUBPROGRAM "bargdes"

Variable	Definition (units)
abest:	Optimum frame area (includes web area and flange area) (sq in)
absfact:	File containing data for coeff array
area:	Total longitudinal sectional area of midship crossection (includes longitudinal frames and plates) (sq in)
area1:	Sectional area of bottom longitudinal frame (sq in)
area2:	Sectional area of bottom transverse girder (sq in)
area3:	Sectional area of side longitudinal frame (sq in)
area4:	Sectional area of side transverse girder (sq in)
area5:	Sectional area of deck longitudinal frame (sq in)
area6:	Sectional area of deck transverse girder (sq in)
area7:	Sectional area of bulkhead longitudinal frame (sq in)
area8:	Sectional area of bulkhead transverse girder (sq in)
area9:	Sectional area of double bottom longitudinal frame (sq in)
astan:	Stanchion cross sectional area (sq in)
b:	Molded breadth of barge (ft)
bestlonspac:	Longitudinal frame spacing for minimum hull weight (ft)

bestmodbot: Bottom section modulus for minimum hull weight

bestmoddk: Deck section modulus for minimum hull weight

bestn: Value of n for minimum hull weight

bestxvrspac: Transverse girder spacing for minimum hull weight

bhdplt: Thickness of bulkhead plate (in)

botplt: Thickness of bottom plate (in)

cb: Block coefficient of barge

cendbgirdth: Thickness of double bottom center girder (in)

centroid: Centroid of longitudinal midship crossection area

coeff: Array containing ABS factors used in section modulus calculation

countr: Incremental counter used in do loops

cycle: Index indicating which type of output is desired. If 0, a complete summary of hull scantlings is provided. If 1, only the hull weight and hull section modulus is provided for each case of transverse and longitudinal spacing and value of n. If 2, only the minimum hull weight for all the iterations is provided.

d: Molded depth of barge (ft)

dbdep: Depth of double bottom (in)

dbplt: Thickness of double bottom plate (in)

deepl: Depth of bottom longitudinal frame (in)

deep2: Depth of bottom transverse girder (in)

deep3: Depth of side longitudinal frame (in)

deep5: Depth of deck longitudinal frame (in)

deep6: Depth of deck transverse girder (in)

deep7: Depth of bulkhead longitudinal frame (in)

dimen: See attached sheet

dkplt: Thickness of deck plate (in)

effplate1: Effective plate width to be used in section modulus calculation of bottom longitudinals (ft)

effplate2: Effective plate width to be used in section modulus calculation of bottom transverses (ft)

effplate3: Effective plate width to be used in section modulus calculation of side longitudinals (ft)

effplate4: Effective plate width to be used in section modulus calculation of side transverses (ft)

effplate5: Effective plate width to be used in section modulus calculation of deck longitudinals (ft)

effplate6: Effective plate width to be used in section modulus calculation of deck transverses (ft)

effplate7: Effective plate width to be used in section modulus calculation of bulkhead longitudinals (ft)

effplate8: Effective plate width to be used in section modulus calculation of bulkhead transverses (ft)

effplate9: Effective plate width to be used in section modulus calculation of double bottom longitudinals (ft)

fact: ABS factor used in hull section modulus calculations

fbest: Optimum flange depth of member (in)

flanth1: Thickness of flange of bottom longitudinal frame (in)

flanth2: Thickness of flange of bottom transverse girder (in)

flanth3: Thickness of flange of side longitudinal frame (in)

flanth4: Thickness of flange of side transverse girder (in)

flanth5: Thickness of flange of deck longitudinal frame (in)

flanth6: Thickness of flange of deck transverse girder (in)

flanth7: Thickness of flange of bulkhead longitudinal frame (in)

flanth8: Thickness of flange of bulkhead transverse girder (in)

flanth9: Thickness of flange of double bottom longitudinal frame (in)

flanwil: Depth of flange of bottom longitudinal frame (in)

flanwi2: Depth of flange of bottom transverse girder (in)

flanwi3: Depth of flange of side longitudinal frame (in)

flanwi4: Depth of flange of side transverse girder (in)

flanwi5: Depth of flange of deck longitudinal frame (in)

flanwi6: Depth of flange of deck transverse girder (in)

flanwi7: Depth of flange of bulkhead longitudinal frame (in)

flanwi8: Depth of flange of bulkhead transverse girder (in)

flanwi9: Depth of flange of double bottom longitudinal frame (in)

ftwt: Weight per longitudinal foot of midship section (lb/ft)

gtype: Index used to indicate in smact2 type of transverse girder for which section modulus is being calculated (1--bottom; 2--side; 3--deck; 4--bulkhead).

h1: Head to be used for bottom longitudinal frame/ transverse girder section modulus calculation (ft)

h2: Head to be used for side and longitudinal bulkhead longitudinal frame/transverse girder section modulus calculation (ft)

h3: Head to be used for deck longitudinal frame/ transverse girder section modulus calculation (ft)

h5: Head to be used for double bottom longitudinal frame section modulus calculation (ft)

i: Index used in do loops

ihc: Index used to indicate if deck/bottom longitudinal and transverse section modulus calculation is being made for the first time (ihc=0) or is called for by statements in hull section modulus calculation (ihc=1) which results from bottom/deck plate increases

incrlonspac: Longitudinal spacing increment used in do loops

incrxvrspac: Transverse spacing increment used in do loops

inertia: Moment of inertia of longitudinal midsection area

k1, k2, k3: Coefficients used in calculation of astan

l: Length between perpendiculars of barge (ft)

l2: Unsupported span of deck and bottom transverse girders (ft)

l4: Unsupported span of side and bulkhead transverse girders (ft)

longarea: Sectional area of longitudinals and plates in midship cross-section (sq in)

longwt: Weight per foot for longitudinals and plates in midship cross-section (lbs/foot)

lonspac: Spacing between longitudinals (ft)

maxlonspac: Maximum longitudinal spacing to be considered in the run (ft)

maxn: Upper bound of iterative loop for n

maxxvrspac: Maximum transverse spacing to be considered in the run (ft)

minn: Lower bound of iterative loop for n

minlonspac: Minimum longitudinal spacing to be considered in

the run (ft)

minwt: The minimum hull weight calculated for all variations in longitudinal and transverse spacing and value of n (LT)

minxvrspac: Minimum transverse spacing to be considered in the run (ft)

modbot: Calculated section modulus to bottom plate (hull girder section modulus)

modbottemp: Temporary storage variable for bottom section modulus used in reducing iterations of n values

moddk: Calculated section modulus to deck plate (hull girder section modulus)

moddktemp: Temporary storage variable for deck section modulus used in reducing interations of n values

modval1: Calculated section modulus of bottom longitudinal frame

modval2: Calculated section modulus of bottom transverse girder

modval3: Calculated section modulus of side longitudinal frame

modval4: Calculated section modulus of side transverse girder

modval5: Calculated section modulus of deck longitudinal frame

modval6: Calculated section modulus of deck transverse girder

modval7: Calculated section modulus of bulkhead longitudinal frame

modval8: Calculated section modulus of bulkhead transverse girder

modval9: Calculated section modulus of double bottom longitudinal frame

ntrbkd: Number of transverses between transverse bulkheads

n: Index used to indicate how many 1/16's inch deckplate may exceed bottom plate in hull section modulus calculations

opt: Index used to indicate which longitudinal section option is being considered. If opt=0 then deck/bottom transverse span is limited to less than 15'. If opt=1, then transverse span is increased by reducing number of stanchions by two in transverse cross-section. If opt=2 then two longitudinal bulkheads are used, with no stanchions. If opt=3 then opt=0 applies but with double bottom.

scanlim: Switch to indicate scantlings limit reached in smact1 or smact2 (scanlim = 1 if scantling limits exceeded to obtain required section modulus).

secmod1: Required section modulus of bottom longitudinal frame

secmod2: Required section modulus of bottom transverse girder

secmod3: Required section modulus of side longitudinal frame

secmod4: Required section modulus of side transverse girder

secmod5: Required section modulus of deck longitudinal frame

secmod6: Required section modulus of deck transverse girder

secmod7: Required section modulus of bulkhead longitudinal frame

secmod8: Required section modulus of bulkhead transverse girder

secmod9: Required section modulus of double bottom longitudinal frame

siddbgirdth: Thickness of double bottom side girders (in)

sidplt: Thickness of side shell plating (in)

sigm_a: Sum of longitudinal member (longitudinal frames and plates) cross-sectional areas (sq in)

sigma_ay: Sum of first vertical moments about baseline (ft-in^2)

sigma_ay2: Sum of second vertical moments about baseline (ft-in)^2

sigma_io: Sum of longitudinal member individual inertias about their respective center of gravity (in^4)

smbest: Optimum section modulus of member

smreq: Required hull girder section modulus

stans: The number of stanchions used in supporting a deck/bottom transverse girder

t: Loaded draft of barge (ft)

tfbest: Optimum flange thickness of member (in)

twbest: Optimum web thickness of member (in)

wbest: Optimum web depth of member (in)

webdep1: Depth of the web of bottom longitudinal frame (in)

webdep2: Depth of the web of bottom transverse girder (in)

webdep3: Depth of the web of side longitudinal frame (in)

webdep4: Depth of the web of side transverse girder (in)

webdep5: Depth of the web of deck longitudinal frame (in)

webdep6: Depth of the web of deck transverse girder (in)

webdep7: Depth of the web of bulkhead longitudinal frame (in)

webdep8: depth of the web of bulkhead transverse girder (in)

webdep9: Depth of the web of double bottom longitudinal frame (in)

webth1: Thickness of the web of bottom longitudinal frame (in)

webth2: Thickness of the web of bottom transverse girder (in)

webth3: Thickness of the web of side longitudinal frame (in)

webth4: Thickness of the web of side transverse girder (in)

webth5: Thickness of the web of deck longitudinal frame (in)

webth6: Thickness of the web of deck transverse girder (in)

webth7: Thickness of the web of bulkhead longitudinal frame (in)

webth8: Thickness of the web of bulkhead transverse girder (in)

webth9: Thickness of the web of double bottom longitudinal frame (in)

weight: Estimated hull steel weight of barge (LT)

wtstan: Weight per stanchion (lbs)

xvrs1: Weight per longitudinal foot for bottom transverse girders (lbs/ft)

xvrs2: Weight per longitudinal foot for side transverse girders (lbs/ft)

xvrs3: Weight per longitudinal foot for deck transverse girders (lbs/ft)

xvrs4: Weight per longitudinal foot for bulkhead transverse girders (lbs/ft)

xvrs5: Weight per longitudinal foot for transverse bulkheads (lbs/ft)

xvrs6: Weight per longitudinal foot for transverse bulkhead longitudinals (lbs/ft)

xvrs7: Weight per longitudinal foot for stanchions (lbs/ft)

xvrspac: Spacing between transverse girders (ft)

DEFINITION OF THE ARRAY DIMEN

Dimensions	Member	Definition
(1,1)	Bottom Plate	12xb
(1,2)	" "	botplt
(1,3)	" "	$\frac{\text{botplt}}{24}$
(2,1)	Bottom Longitudinal Web	$\frac{\text{webth1xb}}{\text{lonspac}}$
(2,2)	" " "	webdep1
(2,3)	" " "	$\frac{\text{botplt}+\frac{\text{webdep1}}{2}}{12}$
(3,1)	Bottom Longitudinal Flange	$\frac{\text{flanwi1 x b}}{\text{lonspac}}$
(3,2)	" " "	flanth1
(3,3)	" " "	$\frac{\text{botplt}+\text{webdep1}+\frac{\text{flanth1}}{2}}{12}$
(4,1)	Side Plate	2 x sidplt
(4,2)	" "	12 x d
(4,3)	" "	$\frac{d}{2}$
(5,1)	Side Longitudinal Web	$\frac{\text{webdep3x2xd}}{\text{lonspac}}$
(5,2)	" " "	webth3
(5,3)	" " "	$\frac{d}{2}$
(6,1)	Side Longitudinal Flange	$\frac{\text{flanth3x2xd}}{\text{lonspac}}$
(6,2)	" " "	flanwi3
(6,3)	" " "	$\frac{d}{2}$

Dimensions	Member	Definition
(7,1)	Deck Plate	12xb
(7,2)	" "	dkplt
(7,3)	" "	$d-\frac{dkplt}{24}$
(8,1)	Deck Longitudinal	$\frac{webth5xb}{lonspac}$
(8,2)	" "	webdep5
(8,3)	" "	$\frac{d-(dkplt+\frac{webdep5}{2})}{12}$
(9,1)	Deck Longitudinal Flange	$\frac{flanwi5xb}{longspac}$
(9,2)	" "	flanth5
(9,3)	" "	$\frac{d-dkplt+webdep5+\frac{flanth5}{2}}{12}$
(10,1)	Bulkhead Plate	bhdplt
(10,2)	" "	12xd - dbdep
(10,3)	" "	$\frac{d}{2}+\frac{dbdep}{24}$
(11,1)	Bulkhead Longitudinal Web	$\frac{webdep7x(d-\frac{dbdep}{12})}{lonspac}$
(11,2)	" " "	webth7
(11,3)	" " "	$\frac{d}{2}+\frac{dbdep}{24}$
(12,1)	Bulkhead Longitudinal Flange	$\frac{flanth7xd}{lonspac}$
(12,2)	" "	flanwi7
(12,3)	" "	$\frac{d}{2}+\frac{dbdep}{24}$

Dimensions	Member	Definition
(13,1)	Double Bottom Plate	$12 \times b$
(13,2)	" " "	dbplt
(13,3)	" " "	$\dfrac{\text{botplt+dbdep+}\frac{\text{dbdplt}}{2}}{12}$
(14,1)	Double Bottom Longitudinal Web	$\dfrac{\text{webth9xb}}{\text{lonspac}}$
(14,2)	" " "	webdep9
(14,3)	" " "	$\dfrac{\text{botplt+dbdep-}\frac{\text{web9}}{2}}{12}$
(15,1)	Double Bottom Longitudinal Flange	$\dfrac{\text{flanwi9xb}}{\text{lonspac}}$
(15,2)	" " "	flanth9
(15,3)	" " "	$\dfrac{\text{botplt+dbdep-webdep9-}\frac{\text{flanth9}}{2}}{12}$
(16,1)	Double Bottom Center And Side Girders	$\text{2xsiddbgirdthx(ceil(}\frac{b}{30}\text{)-1)} + \text{cendbgirdth}$
(16,2)	" " "	dbdep
(16,3)	" " "	$\dfrac{\text{botplt + }\frac{\text{dbdep}}{2}}{12}$

dimen (i,1) ← horizontal dimension of member i

dimen (i,2) ← vertical dimension of member i

dimen (i,3) ← vertical distance of center of area of member i from baseline (bottom of bottom plate)

DEFINITION OF VARIABLES USED IN SUBROUTINES "smact1" AND "smact2"

af: Cross sectional area of flange (sq in)

am: Cross sectional area of member--flange and web (sq in)

ap: Cross section area of plate that is considered effective in section modulus calculation (sq in)

aw: Cross sectional area of web (sq in)

dep: Subroutine input parameter used to specify longitudinal depth in associated transverse girder (in)

f: Flange depth (in)

im: Moment of inertia of member

p: Longitudinal spacing (in)

s: Subroutine input parameter used to specify effective width of plate (ft)

smf: Section modulus of member to the flange

smp: Section modulus of member to the plate

smr: Subroutine input parameter used to specify required section modulus of member

tf: Flange thickness (in))

tp: Subroutine input parameter used to specify plate thickness (in)

tw: Web thickness (in)

w: Web depth (in)

yf: Vertical distance from plate outer surface to flange center of gravity (in)

ym: Vertical distance from plate outer surface to member center of gravity (in)

yp: Vertical distance from plate outer surface to plate center of gravity (in)

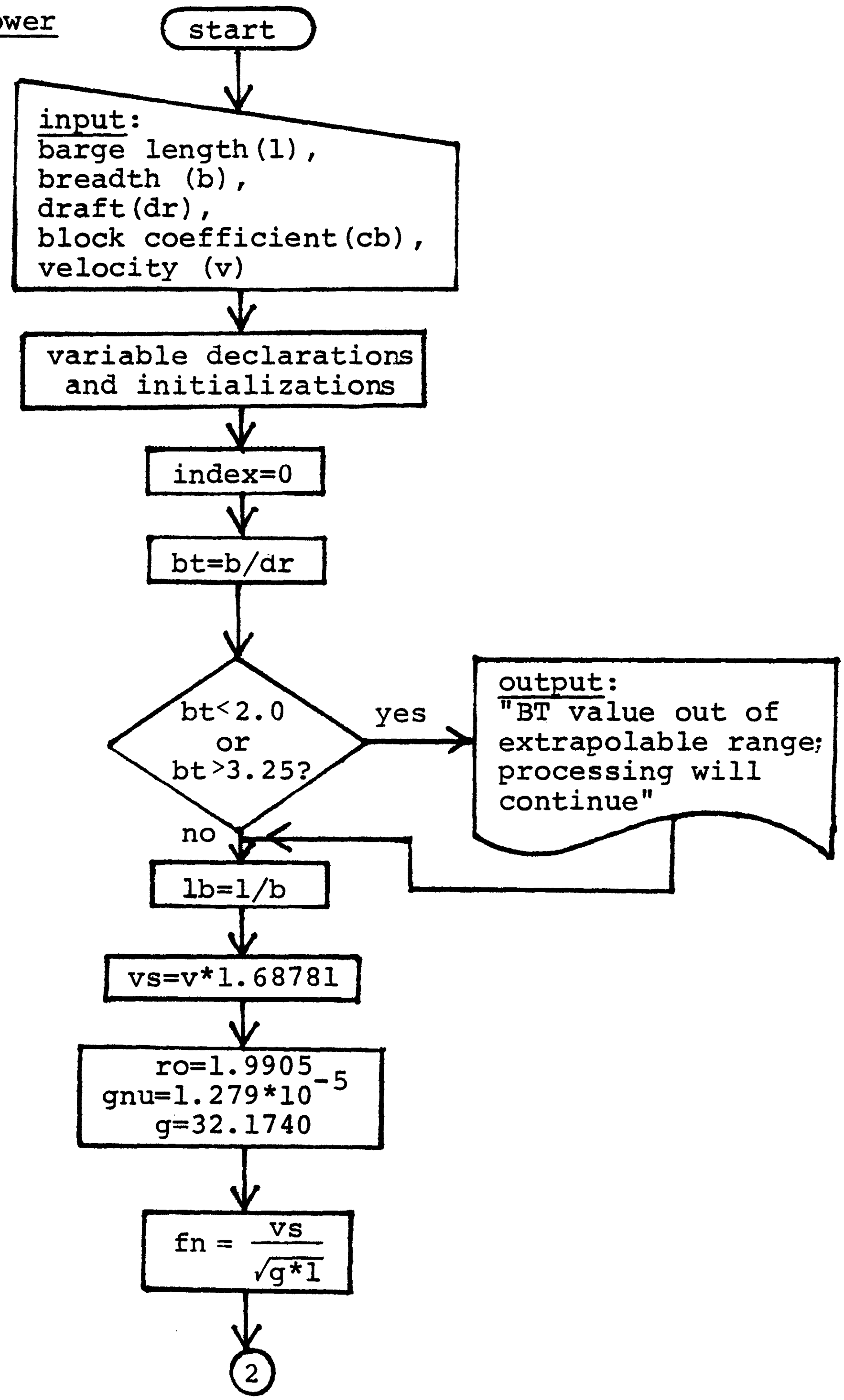
yw: Vertical distance from plate outer surface to web center of gravity (in)

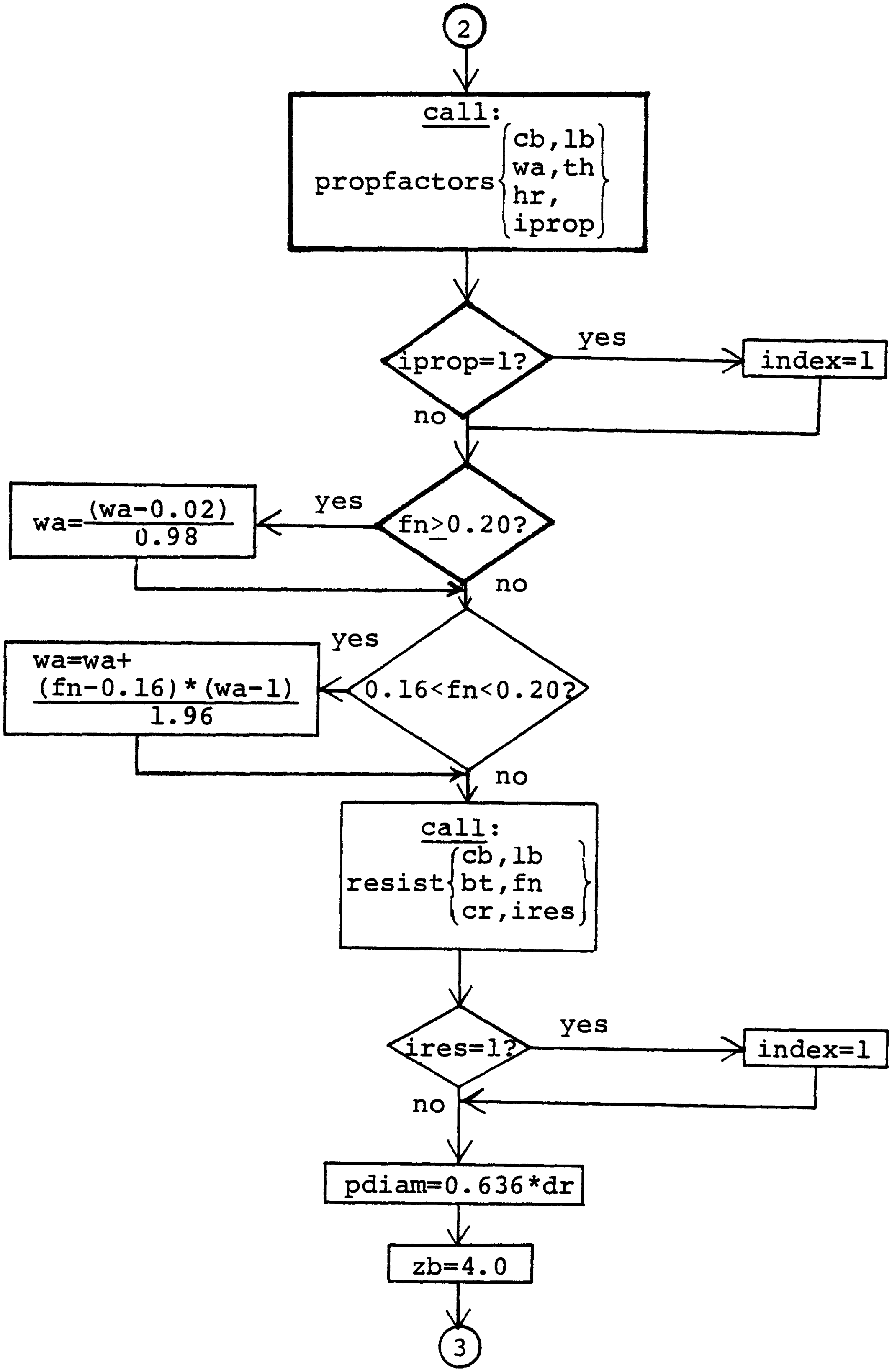
APPENDIX C

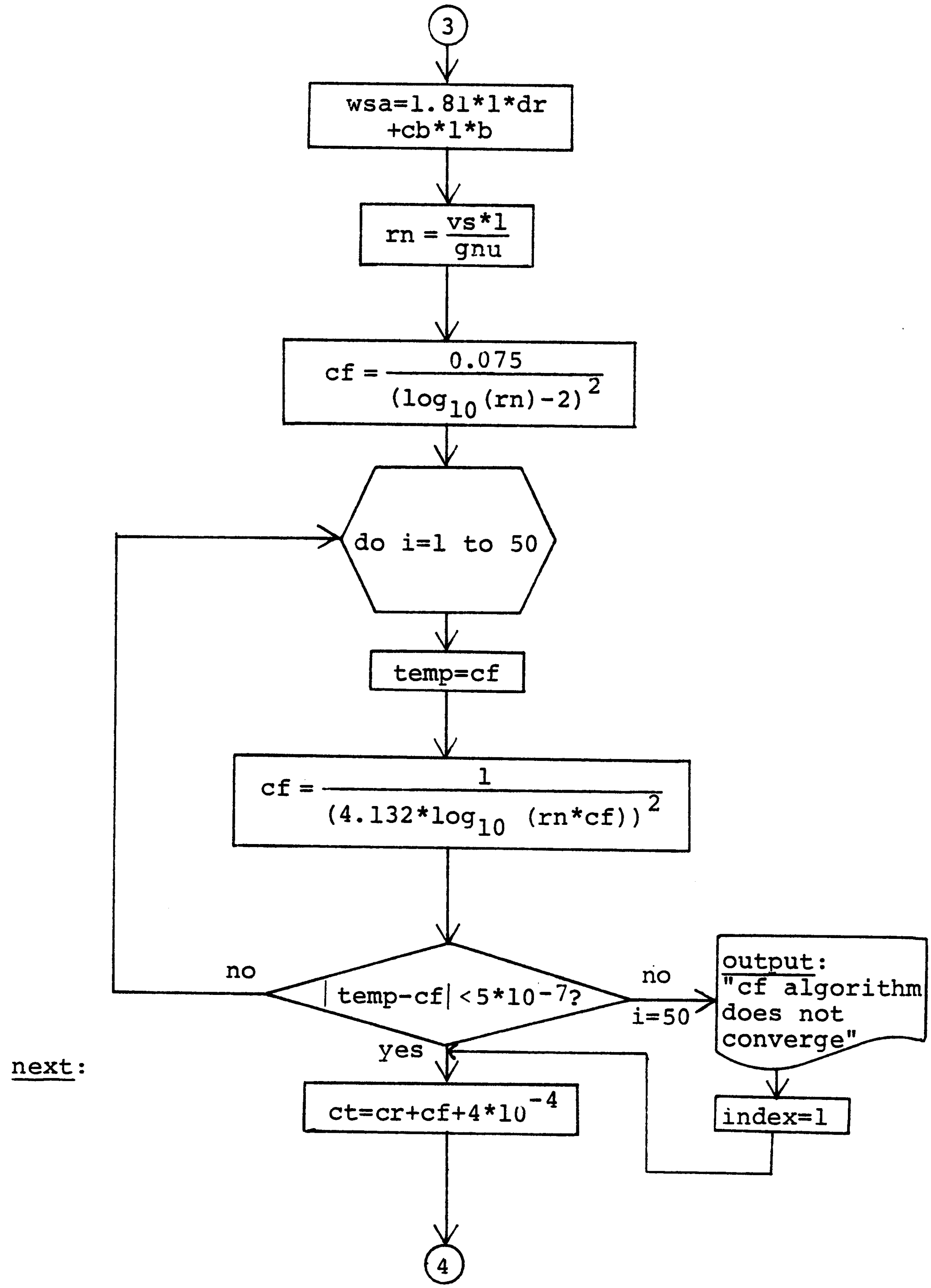
DOCUMENTATION FOR THE BARGE POWERING MODEL

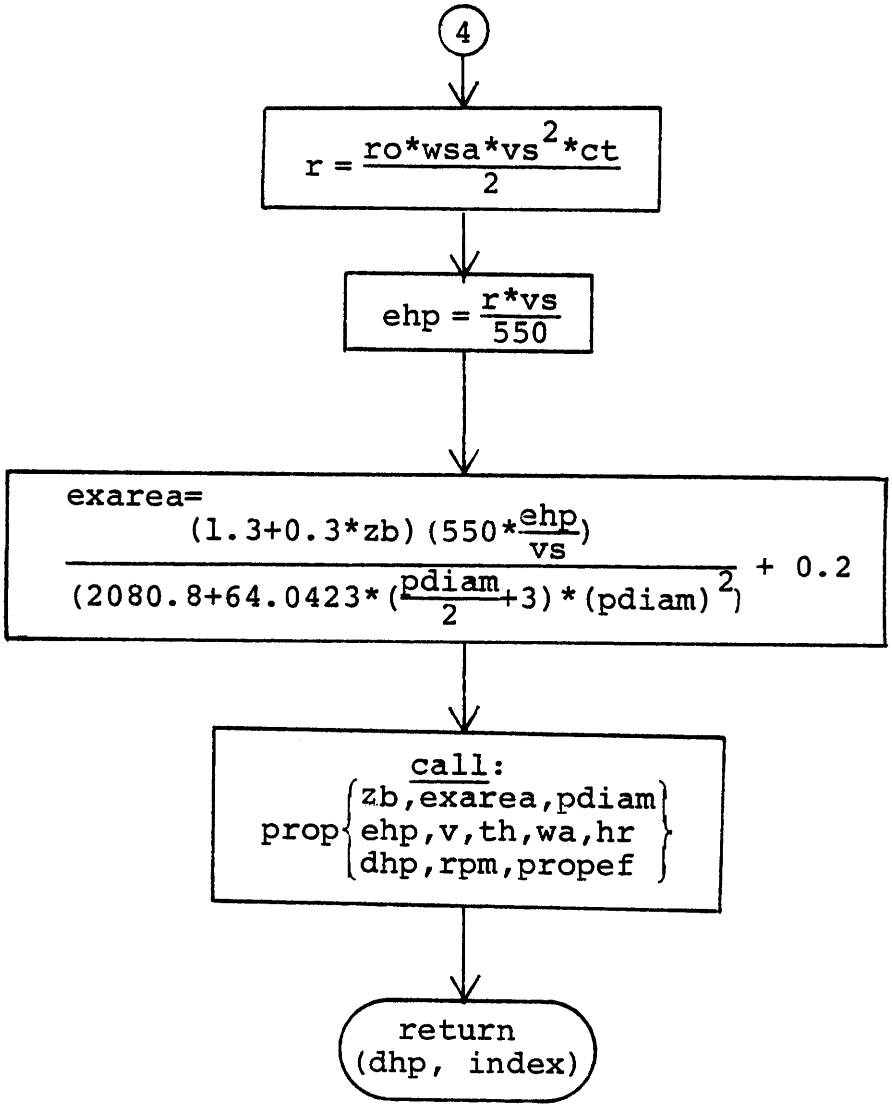
C.1 FLOWCHART OF THE BARGE POWERING MODEL

Subprogram: Power

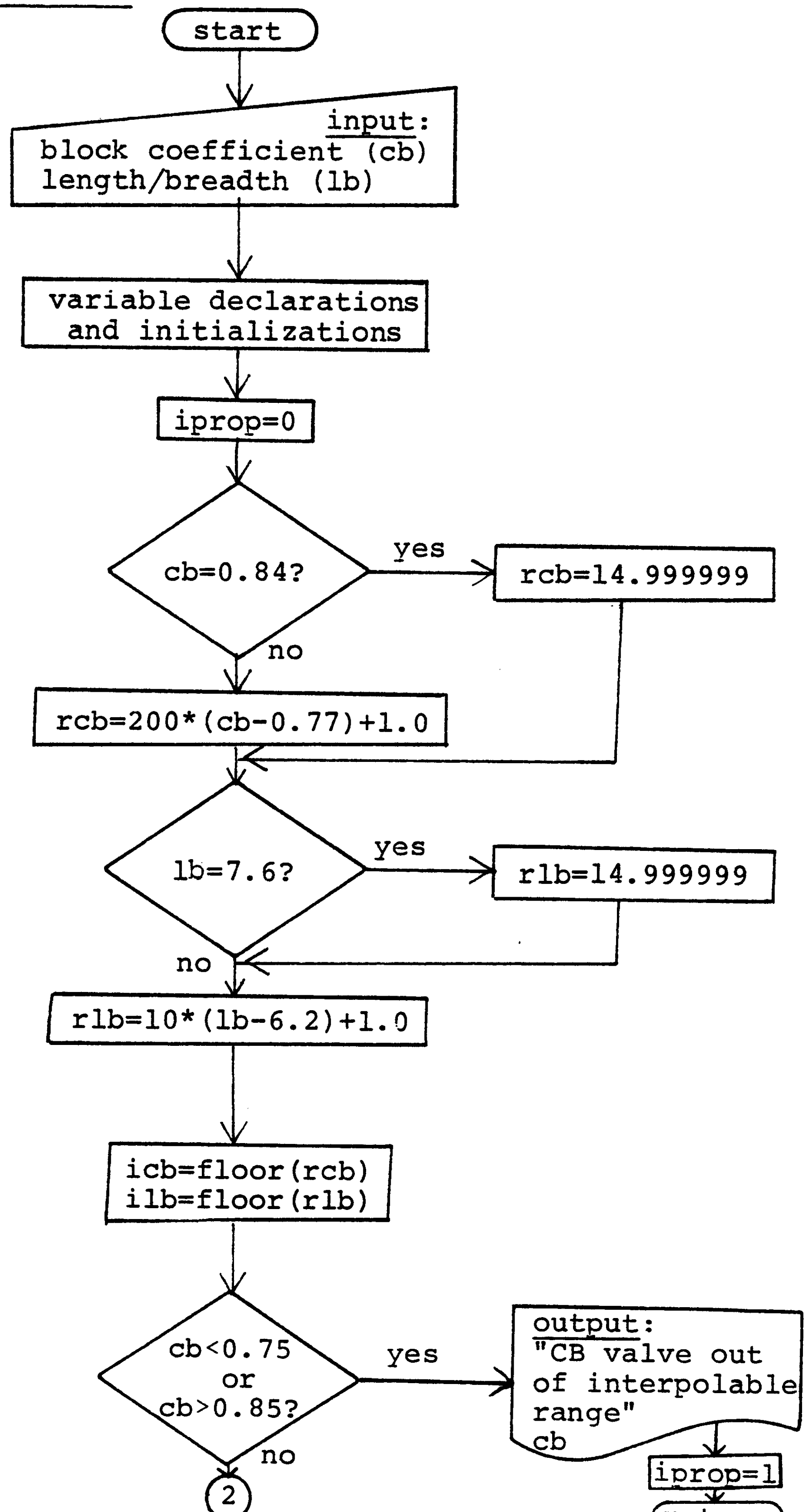


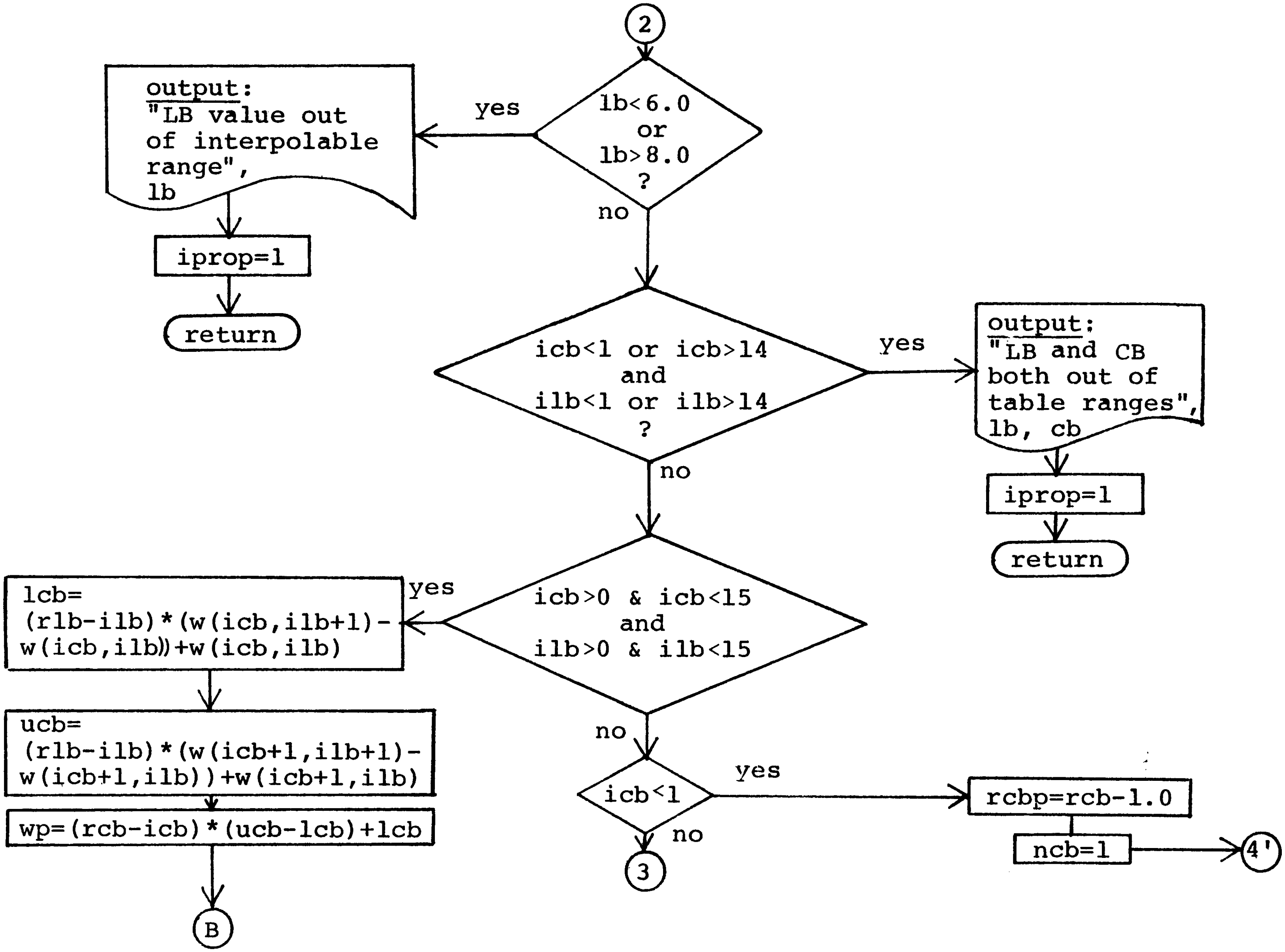

2
call:
propfactors cb,lb wa,th hr, iprop
iprop=1?
yes
index=1
no
fn≥0.20?
yes
wa=(wa-0.02)/0.98
no
0.16<fn<0.20?
yes
wa=wa+(fn-0.16)*(wa-1)/1.96
no
call:
resist cb,lb bt,fn cr,ires
ires=1?
yes
index=1
no
pdiam=0.636*dr
zb=4.0
3

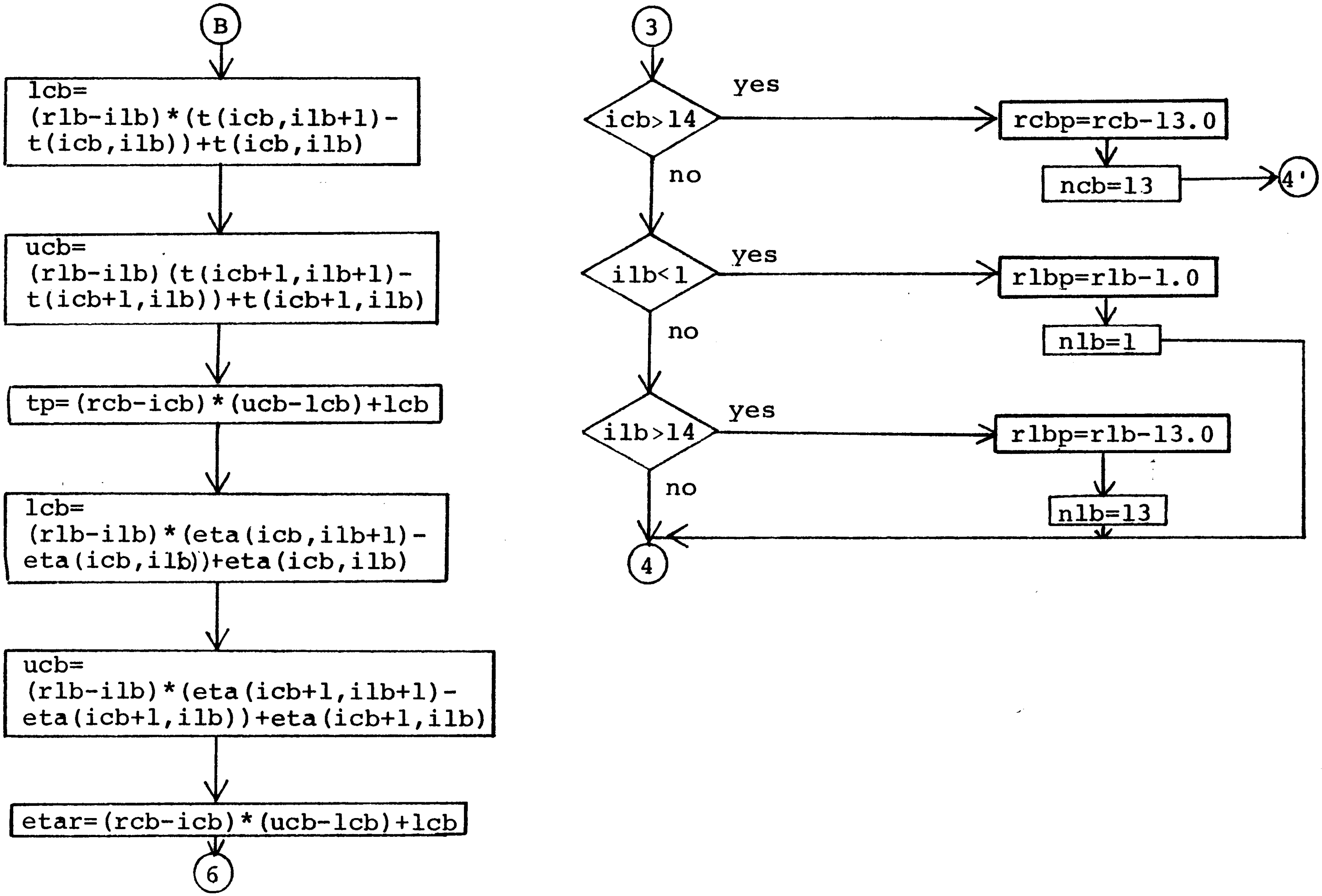

3
wsa=1.81*l*dr
+cb*l*b
rn = vs*l / gnu
cf = 0.075 / (log10(rn)-2)^2
do i=1 to 50
temp=cf
cf = 1 / (4.132*log10 (rn*cf))^2
|temp-cf| < 5*10^-7?
no
no
i=50
output:
"cf algorithm
does not
converge"
yes
next:
index=1
ct=cr+cf+4*10^-4
4

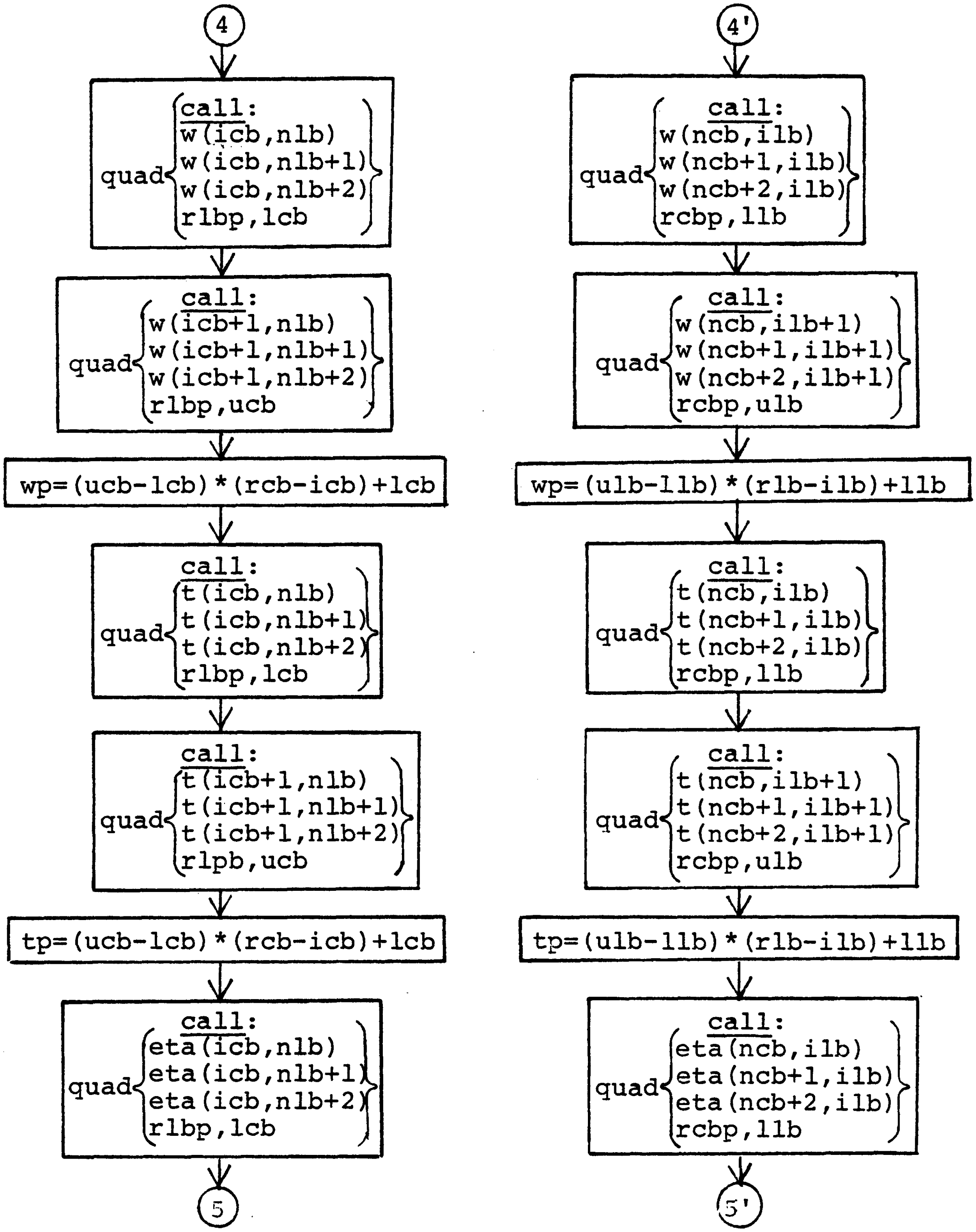

4
r = ro*wsa*vs²*ct / 2
ehp = r*vs / 550
exarea= (1.3+0.3*zb)(550*ehp/vs) / (2080.8+64.0423*(pdiam/2+3)*(pdiam)²) + 0.2
call:
prop {zb,exarea,pdiam ehp,v,th,wa,hr dhp,rpm,propef}
return (dhp, index)

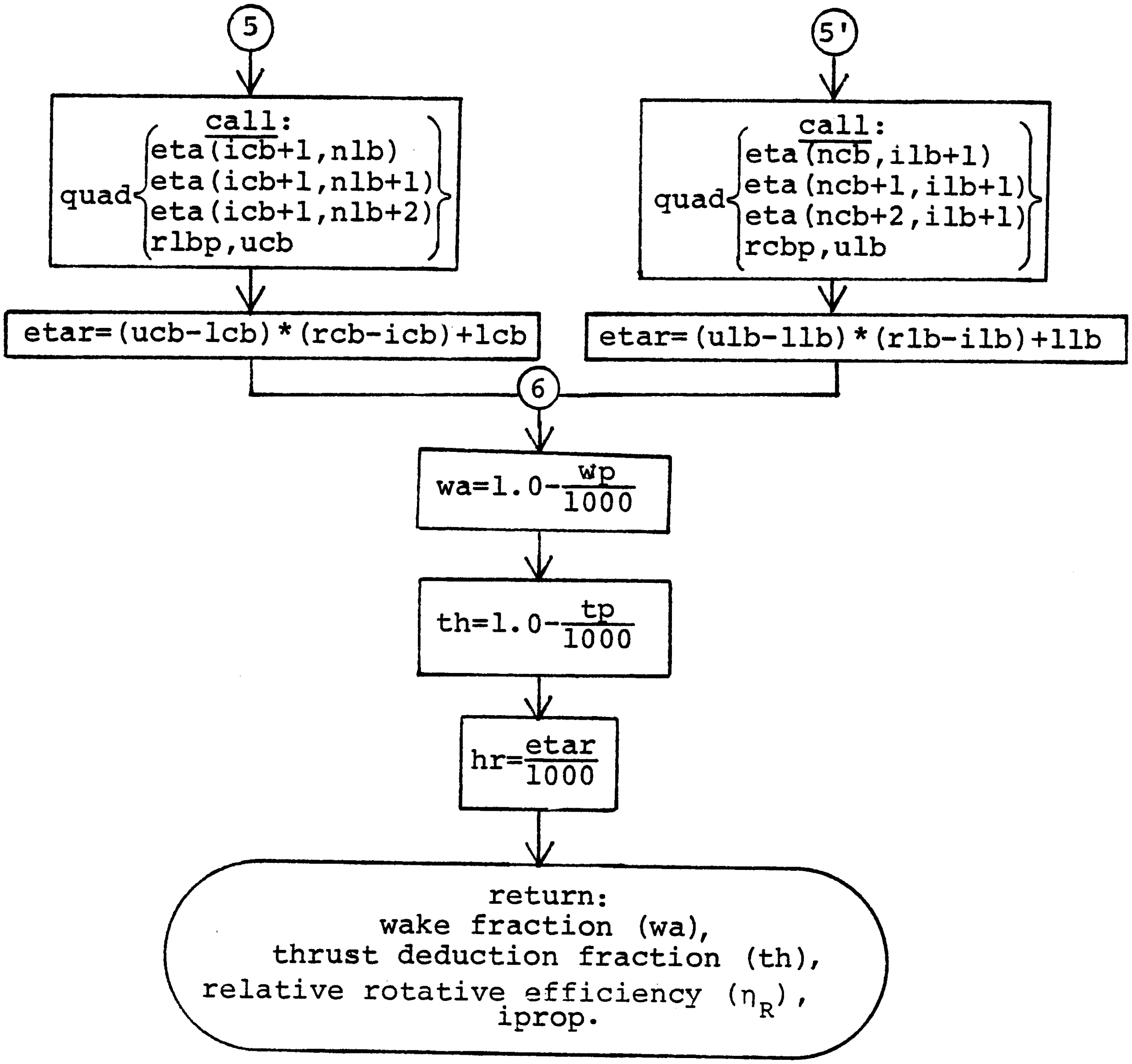
Subroutine Propfactors



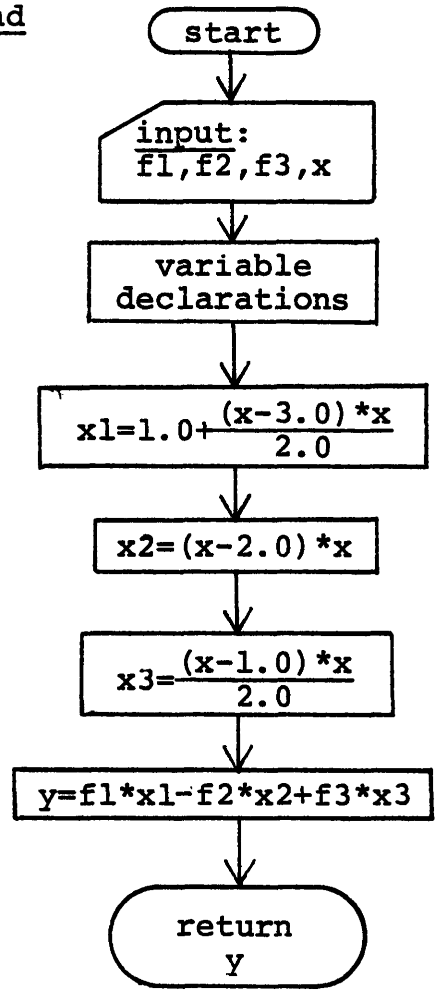

2
lb<6.0 or lb>8.0 ?
yes
output: "LB value out of interpolable range", lb
iprop=1
return
no
icb<1 or icb>14 and ilb<1 or ilb>14 ?
yes
output: "LB and CB both out of table ranges", lb, cb
iprop=1
return
no
icb>0 & icb<15 and ilb>0 & ilb<15
yes
lcb= (rlb-ilb)*(w(icb,ilb+1)-w(icb,ilb))+w(icb,ilb)
ucb= (rlb-ilb)*(w(icb+1,ilb+1)-w(icb+1,ilb))+w(icb+1,ilb)
wp=(rcb-icb)*(ucb-lcb)+lcb
B
no
icb<1
yes
rcbp=rcb-1.0
ncb=1
4'
no
3


B
lcb=
(rlb-ilb)*(t(icb,ilb+1)-
t(icb,ilb))+t(icb,ilb)
ucb=
(rlb-ilb)(t(icb+1,ilb+1)-
t(icb+1,ilb))+t(icb+1,ilb)
tp=(rcb-icb)*(ucb-lcb)+lcb
lcb=
(rlb-ilb)*(eta(icb,ilb+1)-
eta(icb,ilb))+eta(icb,ilb)
ucb=
(rlb-ilb)*(eta(icb+1,ilb+1)-
eta(icb+1,ilb))+eta(icb+1,ilb)
etar=(rcb-icb)*(ucb-lcb)+lcb
6
3
icb>14
yes
rcbp=rcb-13.0
ncb=13
4'
no
ilb<1
yes
rlbp=rlb-1.0
nlb=1
no
ilb>14
yes
rlbp=rlb-13.0
nlb=13
no
4


4
quad call:
w(icb,nlb)
w(icb,nlb+1)
w(icb,nlb+2)
rlbp,lcb
quad call:
w(icb+1,nlb)
w(icb+1,nlb+1)
w(icb+1,nlb+2)
rlbp,ucb
wp=(ucb-lcb)*(rcb-icb)+lcb
quad call:
t(icb,nlb)
t(icb,nlb+1)
t(icb,nlb+2)
rlbp,lcb
quad call:
t(icb+1,nlb)
t(icb+1,nlb+1)
t(icb+1,nlb+2)
rlpb,ucb
tp=(ucb-lcb)*(rcb-icb)+lcb
quad call:
eta(icb,nlb)
eta(icb,nlb+1)
eta(icb,nlb+2)
rlbp,lcb
5
4'
quad call:
w(ncb,ilb)
w(ncb+1,ilb)
w(ncb+2,ilb)
rcbp,llb
quad call:
w(ncb,ilb+1)
w(ncb+1,ilb+1)
w(ncb+2,ilb+1)
rcbp,ulb
wp=(ulb-llb)*(rlb-ilb)+llb
quad call:
t(ncb,ilb)
t(ncb+1,ilb)
t(ncb+2,ilb)
rcbp,llb
quad call:
t(ncb,ilb+1)
t(ncb+1,ilb+1)
t(ncb+2,ilb+1)
rcbp,ulb
tp=(ulb-llb)*(rlb-ilb)+llb
quad call:
eta(ncb,ilb)
eta(ncb+1,ilb)
eta(ncb+2,ilb)
rcbp,llb
5'


5
call:
quad
eta(icb+1,nlb)
eta(icb+1,nlb+1)
eta(icb+1,nlb+2)
rlbp,ucb
etar=(ucb-lcb)*(rcb-icb)+lcb
5'
call:
quad
eta(ncb,ilb+1)
eta(ncb+1,ilb+1)
eta(ncb+2,ilb+1)
rcbp,ulb
etar=(ulb-llb)*(rlb-ilb)+llb
6
wa=1.0-wp/1000
th=1.0-tp/1000
hr=etar/1000
return:
wake fraction (wa),
thrust deduction fraction (th),
relative rotative efficiency (ηR),
iprop.

Subroutine: Quad



Definition of the variablesinvolved in "quad", a subroutine for quadratic interpolation and extrapolation.

f1: value of ordinate associated with an abscissa assumed to be 0.

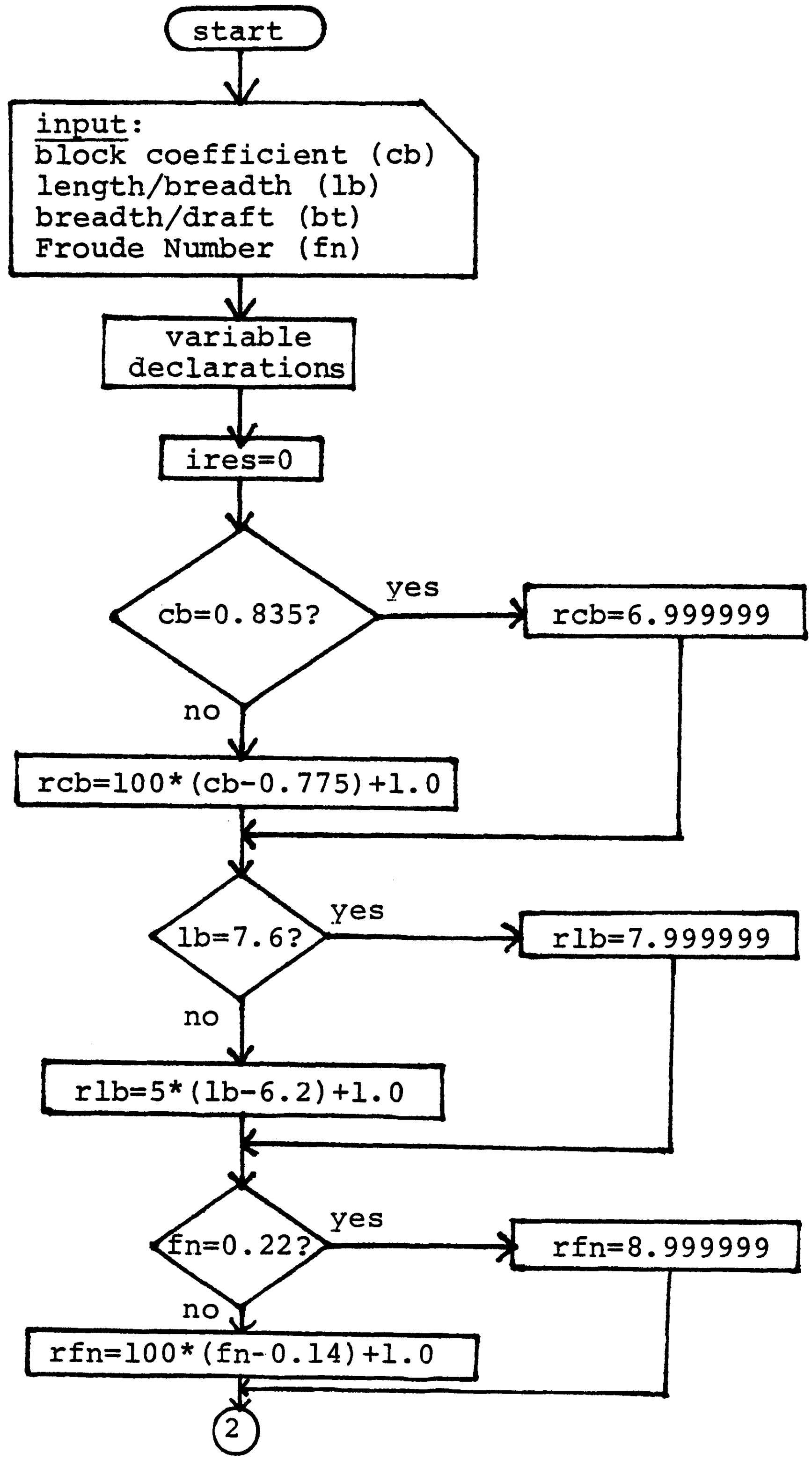
f2: value of ordinate associated with an abscissa assumed to be 1.

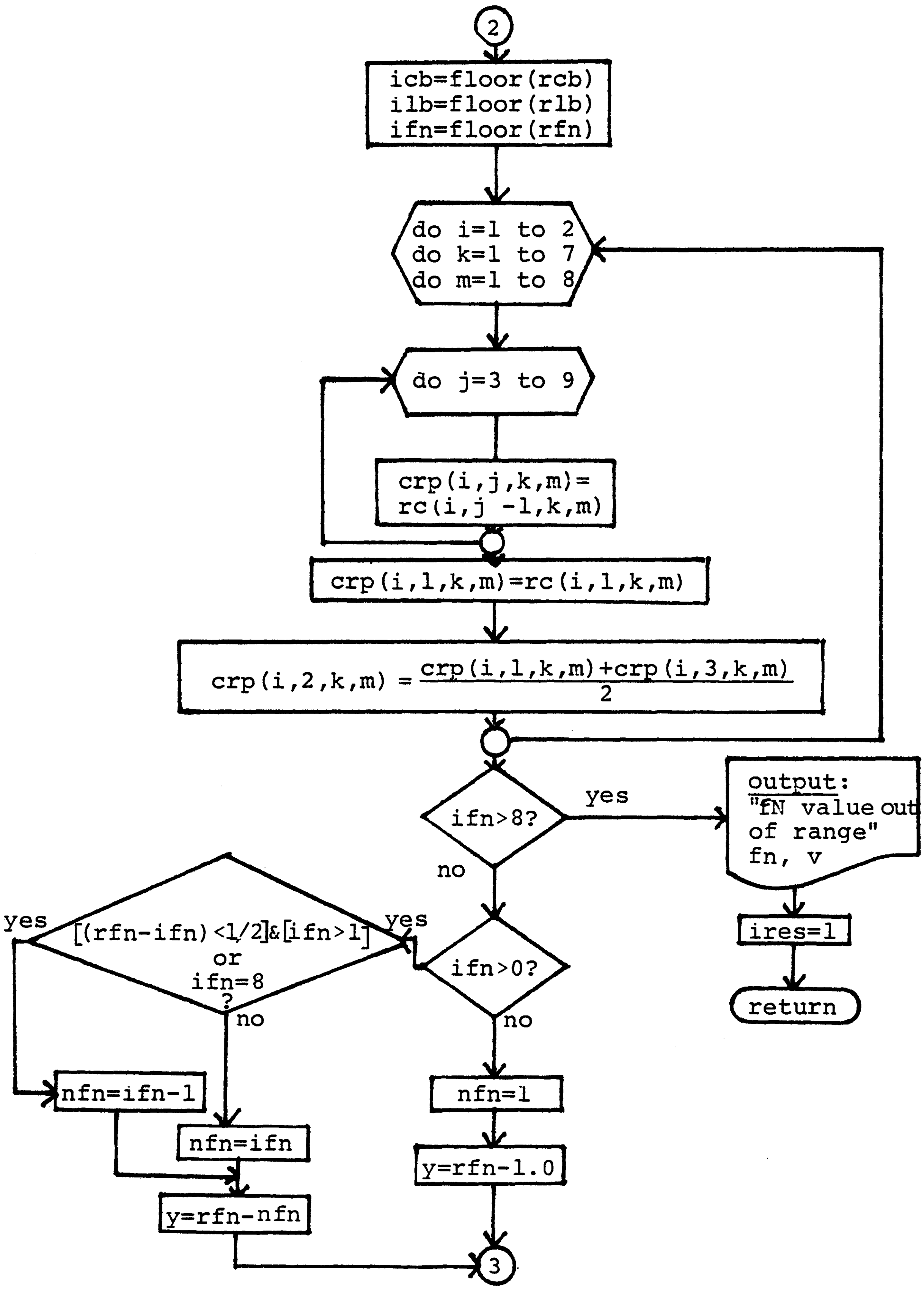
f3: value of ordinate associated with an abscissa assumed to be 2.

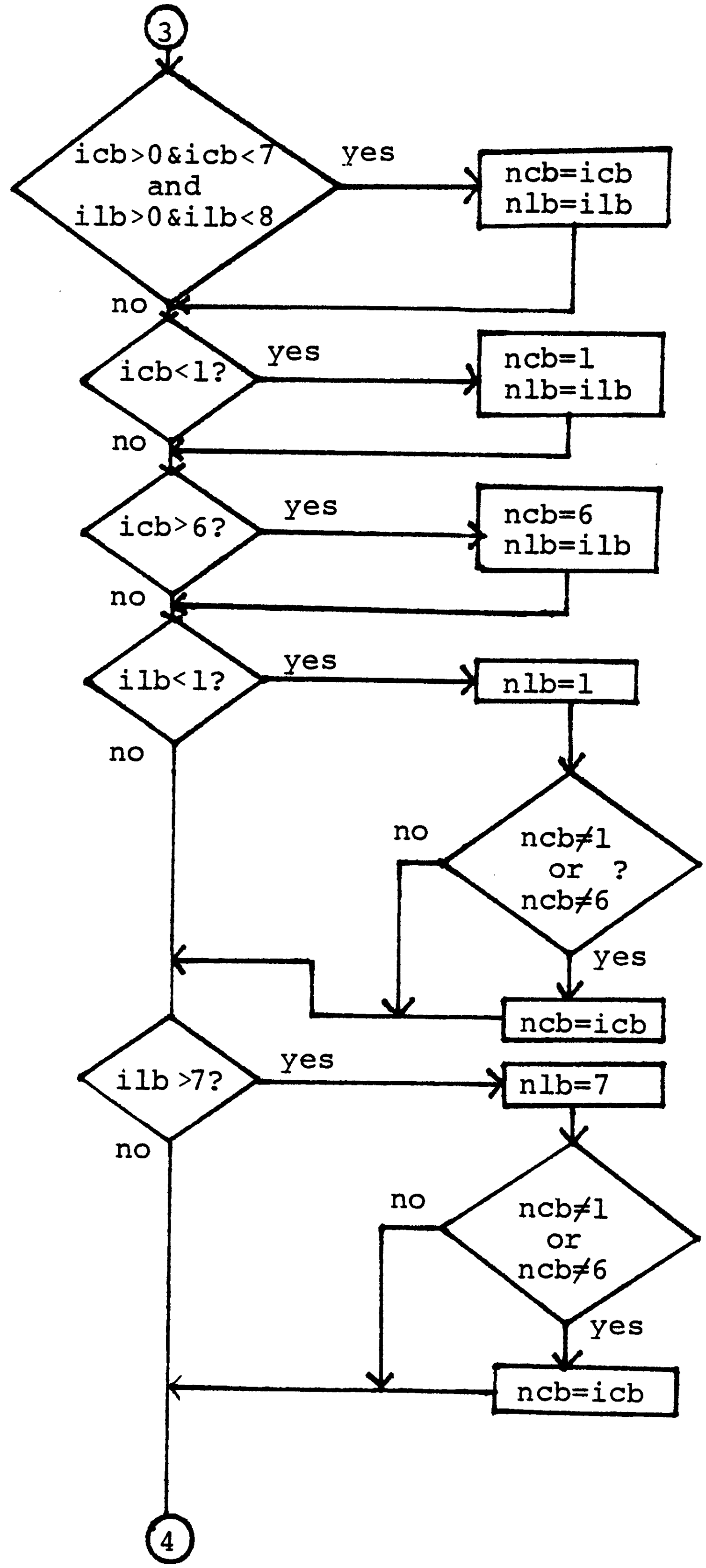
x: value of abscissa for which quadratic interpolation/extrapolation is desired.

y: value of ordinate associated with abscissa x.

Subroutine: resist




2
icb=floor(rcb)
ilb=floor(rlb)
ifn=floor(rfn)
do i=1 to 2
do k=1 to 7
do m=1 to 8
do j=3 to 9
crp(i,j,k,m)=
rc(i,j -1,k,m)
crp(i,1,k,m)=rc(i,1,k,m)
crp(i,2,k,m) = crp(i,1,k,m)+crp(i,3,k,m) / 2
ifn>8?
yes
output:
"fN value out of range"
fn, v
ires=1
return
no
ifn>0?
yes
[(rfn-ifn)<1/2]&[ifn>1]
or
ifn=8
?
yes
no
nfn=ifn-1
nfn=ifn
y=rfn-nfn
no
nfn=1
y=rfn-1.0
3


3
icb>0&icb<7
and
ilb>0&ilb<8
yes
ncb=icb
nlb=ilb
no
icb<1?
yes
ncb=1
nlb=ilb
no
icb>6?
yes
ncb=6
nlb=ilb
no
ilb<1?
yes
nlb=1
no
ncb≠1
or ?
ncb≠6
no
yes
ncb=icb
ilb >7?
yes
nlb=7
no
ncb≠1
or
ncb≠6
no
yes
ncb=icb
4

4

call:

quad
crp(1,nfn,ncb,nlb)
crp(1,nfn+1,ncb,nlb)
crp(1,nfn+2,ncb,nlb)
y,lll

call:

quad
crp(1,nfn,ncb,nlb+1)
crp(1,nfn+1,ncb,nlb+1)
crp(1,nfn+2,ncb,nlb+1)
y,llu

call:

quad
crp(1,nfn,ncb+1,nlb)
crp(1,nfn+1,ncb+1,nlb)
crp(1,nfn+2,ncb+1,nlb)
y,lul

call:

quad
crp(1,nfn,ncb+1,nlb+1)
crp(1,nfn+1,ncb+1,nlb+1)
crp(1,nfn+2,ncb+1,nlb+1)
y,luu

call:

quad
crp(2,nfn,ncb,nlb)
crp(2,nfn+1,ncb,nlb)
crp(2,nfn+2,ncb,nlb)
y,ull

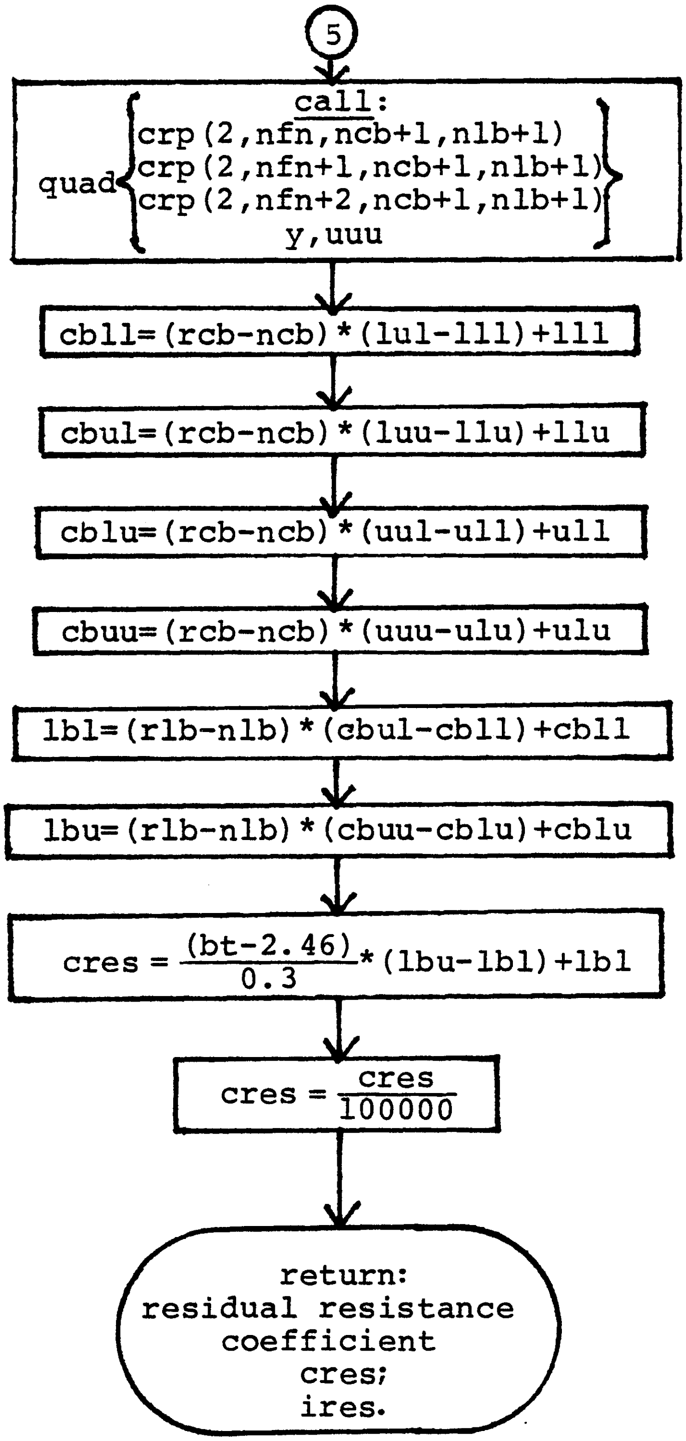
call:

quad
crp(2,nfn,ncb,nlb+1)
crp(2,nfn+1,ncb,nlb+1)
crp(2,nfn+2,ncb,nlb+1)
y,ulu

call:

quad
crp(2,nfn,ncb+1,nlb)
crp(2,nfn+1,ncb+1,nlb)
crp(2,nfn+2,ncb+1,nlb)
y, uul

5


5
call:
quad
crp (2,nfn,ncb+1,nlb+1)
crp (2,nfn+1,ncb+1,nlb+1)
crp (2,nfn+2,ncb+1,nlb+1)
y,uuu
cbll=(rcb-ncb)*(lul-lll)+lll
cbul=(rcb-ncb)*(luu-llu)+llu
cblu=(rcb-ncb)*(uul-ull)+ull
cbuu=(rcb-ncb)*(uuu-ulu)+ulu
lbl=(rlb-nlb)*(cbul-cbll)+cbll
lbu=(rlb-nlb)*(cbuu-cblu)+cblu
cres = (bt-2.46)/0.3 * (lbu-lbl)+lbl
cres = cres/100000
return:
residual resistance
coefficient
cres;
ires.

C.2 COMPUTER PROGRAM LISTING OF THE BARGE POWERING MODEL

```
power: procedure(l,b,dr,cb,v,wa,th,hr,propef,ehp,dhp,index,c,d,dt,dq,rc,t,w,eta);
dcl (c(39),d(47),dt(9),dq(13),rc(2,8,7,8),t(15,15),w(15,15),eta(15,15)) bin float;
dcl (l,b,dr,cb,v,wa,th,hr,propef,ehp,dhp,rpm) bin float;
dcl (cf,cr,ct,temp,r,lb,bt) bin float;
dcl (g,ro,gnu,fn,vs,wsa,rn,exarea,zb,pdiam) bin float;
dcl (i,iprop,ires,index) fixed bin;
dcl sysprint file stream output;
index = 0;
bt = b/dr;
bt = round(bt,2);
if bt<2.0 | bt>3.25 then
     put skip list("BT value out of reasonable extrapolable range--processing will continue");
      lb = l/b;
lb = round(lb,2);
vs = v*1.68781;
g = 32.1740;
ro = 1.9905;
gnu = 1.2791e-5;
fn = vs/sqrt(g*l);
fn = round(fn,3);
call propfactors(cb,lb,wa,th,hr,iprop);
if fn>=0.20 then wa = (wa-0.02)/0.98;
if fn>0.16 & fn<0.20 then wa = wa + (fn-0.16)*(wa-1.0)/1.96;
if iprop = 1 then  do;
     index = 1;
     return;
end;
call resist(cb,lb,bt,fn,cr,ires);
if ires = 1 then  do;
     index = 1;
     return;
end;
pdiam=0.636*dr;
zb = 4.0;
wsa = 1.81*l*dr + cb*l*b;
rn = vs*l/gnu;
cf = 0.075/((log10(rn)-2.0e0)**2);
do i = 1 to 50;
     temp = cf;
     cf = (4.132e0*log10(rn*cf))**-2;
     if abs(temp - cf) < 5.0e-7  then go to next;
end;
put skip list("CF algorithm does not converge");
index = 1;
next:
ct = cr + cf + 0.0004;
r = 0.5*ro*wsa*vs**2*ct;
ehp = r*vs/550.0;
exarea=(1.3 + 0.3*zb)*(550.*ehp/vs)/((2080.8 + 64.0423*(pdiam/2 + 3.0))*pdiam**2) + 0.2;
call prop(zb,exarea,pdiam,ehp,v,th,wa,hr,dhp,rpm,propef);
return;
propfactors: procedure(cb,lb,wa,th,hr,iprop);
dcl (cb,lb,wa,th,hr,wp,tp,etar,rcb,rlb,rcbp,rlbp,lcb,ucb,llb,ulb) bin float;
dcl (iprop,icb,ilb,ncb,nlb) bin fixed;
iprop=0;
if cb=0.84 then rcb=14.999999;  else rcb=(cb-0.77)*200.0 + 1.0;
if lb=7.6 then rlb=14.999999;  else rlb=(lb-6.2)*10.0 + 1.0;
icb=floor(rcb);
ilb=floor(rlb);
if cb<0.75 | cb>0.85 then do;
```

```
     put skip list("CB value out of interpolable range");
     put data(cb);
     iprop=1;
     return;
                         end;
if lb<6.0 ¦ lb>8.0 then do;
     put skip list("LB value out of interpolable range");
     put data(lb);
     iprop=1;
      return;
                         end;
if (icb<2 ¦ icb>14) & (ilb<1 ¦ ilb>14) then do;
/*     put skip list("LB and CB both out table ranges");     */
     iprop=1;
/*     put skip data(lb,cb);      */
     return;
                         end;
if (icb>0 & icb<15) & (ilb>0 & ilb<15) then do;
     lcb=(rlb-ilb)*(w(icb,ilb+1)-w(icb,ilb)) + w(icb,ilb);
     ucb=(rlb-ilb)*(w(icb+1,ilb+1)-w(icb+1,ilb)) + w(icb+1,ilb);
     wp=(rcb-icb)*(ucb-lcb) + lcb;
     lcb=(rlb-ilb)*(t(icb,ilb+1)-t(icb,ilb)) + t(icb,ilb);
     ucb=(rlb-ilb)*(t(icb+1,ilb+1)-t(icb+1,ilb)) + t(icb+1,ilb);
     tp=(rcb-icb)*(ucb-lcb) + lcb;
     lcb=(rlb-ilb)*(eta(icb,ilb+1)-eta(icb,ilb)) + eta(icb,ilb);
     ucb=(rlb-ilb)*(eta(icb+1,ilb+1)-eta(icb+1,ilb)) + eta(icb+1,ilb);
     etar=(rcb-icb)*(ucb-lcb) + lcb;
     go to compl;
                         end;
if icb<1 then do;
     rcbp=rcb-1.0;
     ncb=1;
     go to quad1;
                end;
if icb>14 then do;
     rcbp=rcb-13.0;
     ncb=13;
     go to quad1;
                end;
if ilb<1 then do;
     rlbp=rlb-1.0;
     nlb=1;
     go to quad2;
                end;
if ilb>14 then do;
     rlbp=rlb-13.0;
     nlb=13;
     go to quad2;
                end;
quad2:
call quad(w(icb,nlb),w(icb,nlb+1),w(icb,nlb+2),rlbp,lcb);
call quad(w(icb+1,nlb),w(icb+1,nlb+1),w(icb+1,nlb+2),rlbp,ucb);
wp=(ucb-lcb)*(rcb-icb) + lcb;
call quad(t(icb,nlb),t(icb,nlb+1),t(icb,nlb+2),rlbp,lcb);
call quad(t(icb+1,nlb),t(icb+1,nlb+1),t(icb+1,nlb+2),rlbp,ucb);
tp=(ucb-lcb)*(rcb-icb) + lcb;
call quad(eta(icb,nlb),eta(icb,nlb+1),eta(icb,nlb+2),rlbp,lcb);
call quad(eta(icb+1,nlb),eta(icb+1,nlb+1),eta(icb+1,nlb+2),rlbp,ucb);
etar=(ucb-lcb)*(rcb-icb) + lcb;
go to compl;
```

```
quad1:
call quad(w(ncb,ilb),w(ncb+1,ilb),w(ncb+2,ilb),rcbp,llb);
call quad(w(ncb,ilb+1),w(ncb+1,ilb+1),w(ncb+2,ilb+1),rcbp,ulb);
wp=(ulb-llb)*(rlb-ilb) + llb;
call quad(t(ncb,ilb),t(ncb+1,ilb),t(ncb+2,ilb),rcbp,llb);
call quad(t(ncb, ilb+1),t(ncb+1,ilb+1),t(ncb+2,ilb+1),rcbp,ulb);
tp=(ulb-llb)*(rlb-ilb) + llb;
call quad(eta(ncb,ilb),eta(ncb+1,ilb),eta(ncb+2,ilb),rcbp,llb);
call quad(eta(ncb,ilb+1),eta(ncb+1,ilb+1),eta(ncb+2,ilb+1),rcbp,ulb);
etar=(ulb-llb)*(rlb-ilb) + llb;
compl:
wa=1.0-wp/1000.0;
th=1.0-tp/1000.0;
hr=etar/1000.0;
return;
end propfactors;
quad: procedure(f1,f2,f3,x,y);
dcl (f1,f2,f3,y,x,x1,x2,x3) bin float;
x1=1.0 + (x-3.0)*x/2.0;
x2=(x-2.0)*x;
x3=(x-1.0)*x/2.0;
y=f1*x1 - f2*x2 + f3*x3;
return;
end quad;
resist: procedure(cb,lb,bt,fn,cres,ires);
dcl (cb,lb,bt,fn,crp(2,9,7,8),cres,lll,llu,lul,luu,ull,uul,uuu,ulu,y) float bin;
dcl (rcb,rlb,rfn,cbll,cbul,cblu,cbuu,lbu,lbl) float bin;
dcl (ires,icb,ilb,ifn,nlb,ncb,nfn,i,j,k,m) fixed bin;
ires = 0;
if cb=0.835 then rcb=6.999999;   else rcb=(cb-0.775)*100.0 + 1.0;
if lb=7.6 then rlb=7.999999;   else rlb=(lb-6.2)*5.0 + 1.0;
if fn=0.22 then rfn=8.999999;   else rfn=(fn-0.14)*100.0 + 1.0;
icb=floor(rcb);
ilb=floor(rlb);
ifn=floor(rfn);
do i=1 to 2;
do k=1 to 7;
do m=1 to 8;
do j=3 to 9;
crp(i,j,k,m)=rc(i,j-1,k,m);
end;
crp(i,1,k,m)=rc(i,1,k,m);
crp(i,2,k,m)=0.5*(crp(i,1,k,m) + crp(i,3,k,m));
end;  end;  end;
if ifn>8 then do;
put skip list("FN value out of range");
put skip data(fn,v);
ires=1;
return;
             end;
if ifn>0 then do;
     if ((rfn-ifn)<0.5 & (ifn>1))  | (ifn=8) then nfn=ifn-1;
                                        else nfn=ifn;
     y=rfn-nfn;
             end;
         else do;
             nfn=1;
             y=rfn-1.0;
              end;
if(icb>0 & icb<7) & (ilb>0 & ilb<8) then do;
```

```
      ncb=icb;
      nlb=ilb;
end;
if (icb<1) then do;
      ncb=1;
      nlb=ilb;
end;
if (icb>6) then do;
      ncb=6;
      nlb=ilb;
end;
if (ilb<1) then do;
      nlb=1;
      if ncb^=1 | ncb^=6 then ncb = icb;
end;
if (ilb>7) then do;
      nlb=7;
      if ncb^=1 | ncb^=6 then ncb = icb;
end;
call quad(crp(1,nfn,ncb,nlb),crp(1,nfn+1,ncb,nlb),crp(1,nfn+2,ncb,nlb),y,lll);
call quad(crp(1,nfn,ncb,nlb+1),crp(1,nfn+1,ncb,nlb+1),crp(1,nfn+2,ncb,nlb+1),y,llu);
call quad(crp(1,nfn,ncb+1,nlb),crp(1,nfn+1,ncb+1,nlb),crp(1,nfn+2,ncb+1,nlb),y,lul);
call quad(crp(1,nfn,ncb+1,nlb+1),crp(1,nfn+1,ncb+1,nlb+1),crp(1,nfn+2,ncb+1,nlb+1),y,luu);
call quad(crp(2,nfn,ncb,nlb),crp(2,nfn+1,ncb,nlb),crp(2,nfn+2,ncb,nlb),y,ull);
call quad(crp(2,nfn,ncb,nlb+1),crp(2,nfn+1,ncb,nlb+1),crp(2,nfn+2,ncb,nlb+1),y,ulu);
call quad(crp(2,nfn,ncb+1,nlb),crp(2,nfn+1,ncb+1,nlb),crp(2,nfn+2,ncb+1,nlb),y,uul);
call quad(crp(2,nfn,ncb+1,nlb+1),crp(2,nfn+1,ncb+1,nlb+1),crp(2,nfn+2,ncb+1,nlb+1),y,uuu);
cbll=(rcb-ncb)*(lul-lll) + lll;
cbul=(rcb-ncb)*(luu-llu) + llu;
cblu=(rcb-ncb)*(uul-ull) + ull;
cbuu=(rcb-ncb)*(uuu-ulu) + ulu;
lbl=(rlb-nlb)*(cbul-cbll) + cbll;
lbu=(rlb-nlb)*(cbuu-cblu) + cblu;
cres=((bt-2.46)/0.3)*(lbu-lbl) + lbl;
cres=cres*1.e-5;
return;
end resist;
prop: procedure(zb,x,om,hp,vr,th,wa,hr,dhp,rpm,propef);
       dcl (f1,f2,f3,f4,f5,f6,f7,f8,f9,f10,f11,f12,f13,f14,f15,g1,g2,g3,
      g4,g5,g6,g7,g9,g10,g11,g12,g13,g14,g15,g16,g17,g18,g19,tb1,tb2,
       tb3,tb4,tb5,qb1,qb2,qb3,qb4,qb5,qb6,qb7,tg1,tg2,tg3,tg4,qg1,qg2,qg3,
      qg4,qg5,x,y,fff,om,hp,vr,th,wa,hr,rk,tq1,tq2,tq3,tq4,tq5,tq6,tq7,eps,
       fr1,fr2,fr3,fr4,fr5,fr6,fr7,fr8,gr1,gr2,gr3,y1,y2,ym,
      gr4,gr5,gr6,gr7,gr8,th1,th2,th3,th4,th5,qh1,qh2,qh3,qh4,qh5,dhp,rpm,
      zb,z,z1,z2,zm,t,q,td,qd,tf,qf,propef,aa,bb,cc,for,rn,con,re) bin float;
dcl f(100) bin float;
dcl (k,ll) fixed bin;
eps=0.0001;
       re=1.217*10.**7;
       f1=c(1)+c(8)*zb+c(12)*x*zb+c(18)*x**2+c(29)*zb**2+c(36)*x*zb**2;
       f3=c(7)*x+c(4)+c(28)*x**2*zb;
       f4=c(23)*zb+c(14)+c(26)*x*zb+c(37)*x*zb**2+c(39)*x**2*zb**2;
       f2=c(3)+c(10)*zb;
       f5=c(15);
       f6=c(2)+c(13)*x*zb+c(27)*x**2*zb+c(30)*zb**2;
       f7=c(6)*x+c(11)*zb;
       f8=c(33)*zb**2;
       f9=c(24)*zb;
       f10=c(21)*x**2+c(34)*zb**2;
       f11=c(5)*x+c(9)*zb+c(19)*x**2+c(31)*zb**2;
```

```
      f12=c(16)+c(22)*x**2+c(35)*zb**2;
      f13=c(17)*x+c(20)*x**2+c(32)*zb**2;
      f14=c(25)*zb;
      f15=c(38)*zb**2*x;
      g1=d(1)+d(19)*x**2+d(43)*x**2*zb**2;
      g2=d(5)*x+d(10)*zb+d(22)*x**2;
      g3=d(8)*x+d(4)+d(15)*x*zb+d(33)*x**2*zb+d(41)*x*zb**2;
      g4=d(17)*x+d(25)*x**2+d(44)*x**2*zb**2;
      g5=d(26)*x**2+d(29)*zb+d(31)*x*zb+d(35)*x**2*zb+d(42)*x*zb**2+d(46)
     *x**2*zb**2;
      g6=d(9)*zb+d(20)*x**2+d(32)*x**2*zb+d(39)*x*zb**2;
      g7=d(3)+d(6)*x+d(11)*zb+d(14)*x*zb+d(36)*zb**2;
      g9=d(18)*x;
      g10=d(47)*x**2*zb**2;
      g11=d(2)+d(13)*x*zb+d(40)*x*zb**2;
      g12=d(7)*x+d(12)*zb;
      g13=d(24)*x**2;
      g14=d(34)*x**2*zb;
      g15=d(16)*x+d(21)*x**2+d(27)*zb+d(30)*x*zb;
      g16=d(23)*x**2;
      g17=d(37)*zb**2;
      g18=d(28)*zb+d(45)*zb**2*x**2;
      g19=d(38)*zb**2;
      tb1=dt(3)*x;
      tb2=dt(8)*zb*x;
      tb3=dt(9)*zb**2*x;
      tb4=dt(2)*x;
      tb5=dt(7)*zb*x+dt(4)*x;
      qb1=dq(1)+dq(4)*x**2;
      qb2=dq(12)*x**2;
      qb3=dq(13)*x**2;
      qb4=dq(10)*zb;
      qb5=dq(11)*zb;
      qb6=dq(9)*zb*x;
      qb7=dq(8)*zb*x;
      fff=log10(re)-.301;
      tg1=tb1+tb2*fff;
      tg2=tb3 *fff;
      tg3=tb4+tb5*fff**2;
      tg4=dt(5)*fff             +dt(6)*fff**2;
      qg1=qb1+qb2*fff           +qb3*fff**2;
      qg2=dq(2)+dq(5)*fff;
      qg3=dq(6)*fff           +dq(7)*fff**2;
      qg4=  qb4*fff           +qb5  *fff**2             +dq(3)*zb;
      qg5=qb6*fff**2;
      qh3=qb7*fff;
      th1=dt(1);
   rk=5.2449*hp/((1.-wa)**2*(1.-th)*(om*.3048)**2*vr**3);
      y1=0.60;
      y2=1.40;
      y=y1;
      k=1;
N811:      call grend(for,z);
       f(k)=for;
      k=k+1;
if k<3 then go to N809;
if k>3 then go to N840;
       else go to N810;
  N809: ym=(y1+y2)*.5;
if abs(y1-y2)<=eps*ym then do;
```

```
       y=ym;
       go to zeld;
                           end;
       y=ym;
      go to N811;
  N810: bb=f(1);
  N840: ll=k-1;
      cc=f(ll);
     if bb*cc<0.e0 then y2=ym;
                 else if bb*cc=0.e0 then do;
                              y = ym;
                              go to zeld;
                                             end;
                                      else do;
                                         y1=ym;
                                           bb=cc;
                                             end;
     go to N809;
zeld:       t=fr1+z*fr2+z**2*fr3+z**3*fr4;
      q=gr1+z*gr2+z**2*gr3+z**3*gr4;
      td=th1+z*th2+th3*z**2;
      qd=qh1+z*qh2+z**2*qh3;
      tf=t+td;
      qf=q+qd;
      propef=z*tf/(6.283185308*qf);
      rpm=30.9*vr*(1.-wa)/(z*om*.3048);
      dhp=hp*(1.-wa)/(propef*(1.-th)*hr);
grend: procedure(for,z);
       dcl (for,z) bin float;
      rn=2.;
      fr1=f1+y*f2+y**2*f3+y**6*f5+y**3*f4;
      fr2=f6+y*f7+y**2*f8+y**3*f9+f10*y**6;
      fr3=f11+y**6*f12;
      fr4=f13+f14*y**3+f15*y**6;
      fr5=f2+2.*y*f3+6.*y**5*f5+3.*y**2*f4;
      fr6=f7+2.*y*f8+3.*y**2*f9+6.*y**5*f10;
      fr7=6.*y**5*f12;
      fr8=6.*f15*y**5+3.*f14*y**2;
      gr1=g1+y*g2+y**2*g3+y**3*g4+y**6*g5;
      gr2=g6+y*g7        +g9*y**3+g10*y**6;
      gr3=g11+g12*y+g13*y**2+g14*y**3;
      gr4=g15+g16*y+g17*y**2+g19*y**6+y**3*g18;
      gr5=g2+2.*y*g3+3.*y**2*g4+6.*y**5*g5;
      gr6=g7        +3.*g9*y**2+6.*g10*y**5;
      gr7=g12+2.*g13*y+3.*g14*y**2;
      gr8=g16+2.*g17*y+3.*g18*y**2+6.*g19*y**5;
      th2=y*tg1+y**3*tg2;
      th3=tg3+      y**6*tg4;
      th4=tg1+3.*y**2*tg2;
      th5=6.*y**5*tg4;
      qh1=qg1+y*qg2+y**2*qg3+y**6*qg4;
      qh2=qg5*y;
      qh4=qg2+2.*y*qg3+6.*y**5*qg4;
      qh5=qg5;
      con=rk;
      tq1=fr5*(rn+1.)*(gr1+qh1)-rn*(fr1+th1)*(gr5+qh4);
      tq2=(rn+1.)*(gr1+qh1)*(fr6+th4)+(gr2+qh2)*fr5*rn+(gr5+qh4)*(1.-rn)
     *(fr2+th2)-(gr6+qh5)*    rn *(fr1+th1);
      tq3=fr5*(rn-1.)*(gr3+qh3)+(fr6+th4)*(gr2+qh2)*rn+(fr7+th5)*(rn+1.)
     *(gr1+qh1)+(gr5+qh4)*(2.-rn)*(fr3+th3)+(gr6+qh5)*(1.-rn)*(fr2+th2)
```

```
      -gr7*rn*(fr1+th1);
      tq4=fr5*(rn-2.)*gr4+(fr7+th5)*(gr2+qh2)*rn+(fr6+th4)*(rn-1.)*(gr3+
     qh3)+fr8*(rn+1.)*(gr1+qh1)+(gr5+qh4)*(3.-rn)*fr4+(gr6+qh5)*(2.-rn)
      *(fr3+th3)+gr7*(1.-rn)*(fr2+th2)-gr8*rn*(fr1+th1);
      tq5=(fr6+th4)*(rn-2.)*gr4+(fr7+th5)*(rn-1.)*(gr3+qh3)+fr8*(gr2+qh2)
       *rn+(gr6+qh5)*(3.-rn)*fr4+gr7*(2.-rn)*(fr3+th3)+gr8*(1.-rn)*(fr2+th2);
      tq6=(fr7+th5)*(rn-2.)*gr4+fr8*(rn-1.)*(gr3+qh3)+gr7*(3.-rn)*fr4+gr8
       *(2.-rn)*(fr3+th3);
      tq7=fr8*(rn-2.)*gr4+gr8*(3.-rn)*fr4;
      z1=.01;
      z2=y+.1;
      aa=fr1+th1+(fr2+th2)*z1+(fr3+th3)*z1**2+fr4*
      z1**3-con*z1**2;
  N801: zm=(z1+z2)*.5;
      cc=fr1+th1+(fr2+th2)*zm+(fr3+th3)*zm**2+fr4*
      zm**3-con*zm**2;
if abs(z1-z2)<=eps*zm then go to N805;
if aa*cc=0.e0 then go to N805;
if aa*cc<0.e0 then z2=zm;
           else do;
                  z1=zm;
                 aa=cc;
               end;
go to N801;
N805: z=zm;
      for       =(tq1+z*tq2+z**2*tq3+z**3*tq4+z**4*tq5+z**5*tq6+z**6*tq7)
        *con*z**2/((gr1+qh1+(gr2+qh2)*z+(qh3+gr3)*z**2+gr4*z**3)**2*(fr2
       +th2+2.*(fr3+th3)*z+3.*fr4*z**2 -rn*con*z));
       return;
        end grend;
end prop;
end power;
```

```
\014      SOURCE FILES USED IN THIS COMPILATION.

LINE      NUMBER  DATE MODIFIED     NAME                                PATHNAME
             0    04/14/79  2316.7  power.pl1                           >udd>Barges>Kaskin>power.pl1
\014      NAMES DECLARED IN THIS COMPILATION.

IDENTIFIER               OFFSET     LOC STORAGE CLASS   DATA TYPE              ATTRIBUTES AND REFERENCES

NAMES DECLARED BY DECLARE STATEMENT.
aa                              000364 automatic        float bin(27)          dcl 219 set ref 372 378 379 382
b                                      parameter        float bin(27)          dcl 3 ref 1 9 13 35
bb                              000365 automatic        float bin(27)          dcl 219 set ref 306 309 310 316
bt                                     parameter        float bin(27)          dcl 146 ref 145 214
bt                              000107 automatic        float bin(27)          dcl 4 set ref 9 10 10 11 11 28
c                                      parameter        float bin(27)          array dcl 2 ref 1 230 230 230 230 230 230 231
                                                                               231 231 232 232 232 232 232 233 233 234 235 235
                                                                               235 235 236 236 237 238 239 239 240 240 240 240
                                                                               241 241 241 242 242 242 243 244
cb                                     parameter        float bin(27)          dcl 146 ref 145 150 150
cb                                     parameter        float bin(27)          dcl 53 ref 52 56 56 60 60 62
cb                                     parameter        float bin(27)          dcl 3 set ref 1 21 28 35
cbll                            002074 automatic        float bin(27)          dcl 147 set ref 208 212 212
cblu                            002076 automatic        float bin(27)          dcl 147 set ref 210 213 213
cbul                            002075 automatic        float bin(27)          dcl 147 set ref 209 212
cbuu                            002077 automatic        float bin(27)          dcl 147 set ref 211 213
cc                              000366 automatic        float bin(27)          dcl 219 set ref 308 309 310 316 375 378 379 382
cf                              000101 automatic        float bin(27)          dcl 4 set ref 37 39 40 40 41 45
con                             000371 automatic        float bin(27)          dcl 219 set ref 355 372 375 386 386
cr                              000102 automatic        float bin(27)          dcl 4 set ref 28 45
cres                                   parameter        float bin(27)          dcl 146 set ref 145 214 215 215
crp                             000100 automatic        float bin(27)          array dcl 146 set ref 160 162 163 163 163 200
                                                                               200 200 201 201 201 202 202 202 203 203 203 204
                                                                               204 204 205 205 205 206 206 206 207 207 207
ct                              000103 automatic        float bin(27)          dcl 4 set ref 45 47
d                                      parameter        float bin(27)          array dcl 2 ref 1 245 245 245 246 246 246 247
                                                                               247 247 247 247 248 248 248 249 249 249 249 249
                                                                               249 251 251 251 251 252 252 252 252 252 253 254
                                                                               255 255 255 256 256 257 258 259 259 259 259 260
                                                                               261 262 262 263
dnp                                    parameter        float bin(27)          dcl 3 set ref 1 50
dnp                                    parameter        float bin(27)          dcl 219 set ref 218 327
dq                                     parameter        float bin(27)          array dcl 2 ref 1 269 269 270 271 272 273 274
                                                                               275 282 282 283 283 284
dr                                     parameter        float bin(27)          dcl 3 ref 1 9 33 35
dt                                     parameter        float bin(27)          array dcl 2 ref 1 264 265 266 267 268 268 280
                                                                               280 287
enp                                    parameter        float bin(27)          dcl 3 set ref 1 48 49 50
eps                             000314 automatic        float bin(27)          dcl 219 set ref 228 300 377
eta                                    parameter        float bin(27)          array dcl 2 set ref 1 85 85 85 86 86 86 117 117
                                                                               117 118 118 118 128 128 128 129 129 129
etar                            000102 automatic        float bin(27)          dcl 53 set ref 87 119 130 134
exarea                          000117 automatic        float bin(27)          dcl 5 set ref 49 50
f                               000373 automatic        float bin(27)          array dcl 226 set ref 294 306 308
f1                              000214 automatic        float bin(27)          dcl 219 set ref 230 331
f1                                     parameter        float bin(27)          dcl 138 ref 137 142
f10                             000225 automatic        float bin(27)          dcl 219 set ref 239 332 336
f11                             000226 automatic        float bin(27)          dcl 219 set ref 240 333
f12                             000227 automatic        float bin(27)          dcl 219 set ref 241 333 337
f13                             000230 automatic        float bin(27)          dcl 219 set ref 242 334
f14                             000231 automatic        float bin(27)          dcl 219 set ref 243 334 338
```

f15	000232 automatic	float bin(27)	dcl 219 set ref 244 334 338
f2	parameter	float bin(27)	dcl 138 ref 137 142
f2	000215 automatic	float bin(27)	dcl 219 set ref 233 331 335
f3	000216 automatic	float bin(27)	dcl 219 set ref 231 331 335
f3	parameter	float bin(27)	dcl 138 ref 137 142
f4	000217 automatic	float bin(27)	dcl 219 set ref 232 331 335
f5	000220 automatic	float bin(27)	dcl 219 set ref 234 331 335
f6	000221 automatic	float bin(27)	dcl 219 set ref 235 332
f7	000222 automatic	float bin(27)	dcl 219 set ref 236 332 336
f8	000223 automatic	float bin(27)	dcl 219 set ref 237 332 336
f9	000224 automatic	float bin(27)	dcl 219 set ref 238 332 336
fff	000303 automatic	float bin(27)	dcl 219 set ref 276 277 278 279 280 280 281 281 282 283 283 284 284 285 286
fn	000113 automatic	float bin(27)	dcl 5 set ref 19 20 20 22 23 23 23 28
fn	parameter	float bin(27)	dcl 146 ref 145 152 152 167
for	parameter	float bin(27)	dcl 329 set ref 328 386
for	000367 automatic	float bin(27)	dcl 219 set ref 293 294
fr1	000315 automatic	float bin(27)	dcl 219 set ref 319 331 356 357 359 362 372 375
fr2	000316 automatic	float bin(27)	dcl 219 set ref 319 332 357 359 362 365 372 375 386
fr3	000317 automatic	float bin(27)	dcl 219 set ref 319 333 359 362 365 367 372 375 386
fr4	000320 automatic	float bin(27)	dcl 219 set ref 319 334 362 365 367 369 372 375 386
fr5	000321 automatic	float bin(27)	dcl 219 set ref 335 356 357 359 362
fr6	000322 automatic	float bin(27)	dcl 219 set ref 336 357 359 362 365
fr7	000323 automatic	float bin(27)	dcl 219 set ref 337 359 362 365 367
fr8	000324 automatic	float bin(27)	dcl 219 set ref 338 362 365 367 369
g	000110 automatic	float bin(27)	dcl 5 set ref 16 19
g1	000233 automatic	float bin(27)	dcl 219 set ref 245 339
g10	000243 automatic	float bin(27)	dcl 219 set ref 254 340 344
g11	000244 automatic	float bin(27)	dcl 219 set ref 255 341
g12	000245 automatic	float bin(27)	dcl 219 set ref 256 341 345
g13	000246 automatic	float bin(27)	dcl 219 set ref 257 341 345
g14	000247 automatic	float bin(27)	dcl 219 set ref 258 341 345
g15	000250 automatic	float bin(27)	dcl 219 set ref 259 342
g16	000251 automatic	float bin(27)	dcl 219 set ref 260 342 346
g17	000252 automatic	float bin(27)	dcl 219 set ref 261 342 346
g18	000253 automatic	float bin(27)	dcl 219 set ref 262 342 346
g19	000254 automatic	float bin(27)	dcl 219 set ref 263 342 346
g2	000234 automatic	float bin(27)	dcl 219 set ref 246 339 343
g3	000235 automatic	float bin(27)	dcl 219 set ref 247 339 343
g4	000236 automatic	float bin(27)	dcl 219 set ref 248 339 343
g5	000237 automatic	float bin(27)	dcl 219 set ref 249 339 343
g6	000240 automatic	float bin(27)	dcl 219 set ref 251 340
g7	000241 automatic	float bin(27)	dcl 219 set ref 252 340 344
g8	000242 automatic	float bin(27)	dcl 219 set ref 253 340 344
gnu	000112 automatic	float bin(27)	dcl 5 set ref 18 36
gr1	000325 automatic	float bin(27)	dcl 219 set ref 320 339 356 357 359 362 386
gr2	000326 automatic	float bin(27)	dcl 219 set ref 320 340 357 359 362 365 386
gr3	000327 automatic	float bin(27)	dcl 219 set ref 320 341 359 362 365 367 386
gr4	000333 automatic	float bin(27)	dcl 219 set ref 320 342 362 365 367 369 386
gr5	000334 automatic	float bin(27)	dcl 219 set ref 343 356 357 359 362
gr6	000335 automatic	float bin(27)	dcl 219 set ref 344 357 359 362 365
gr7	000336 automatic	float bin(27)	dcl 219 set ref 345 359 362 365 367
gr8	000337 automatic	float bin(27)	dcl 219 set ref 346 362 365 367 369
hp	parameter	float bin(27)	dcl 219 ref 218 288 327
hr	parameter	float bin(27)	dcl 219 ref 218 327
hr	parameter	float bin(27)	dcl 53 set ref 52 134
hr	parameter	float bin(27)	dcl 3 set ref 1 21 50

i	002110 automatic	fixed bin(17,0)	dcl 148 set ref 156 160 160 162 162 163 163 163
i	000122 automatic	fixed bin(17,0)	dcl 6 set ref 38
icb	002102 automatic	fixed bin(17,0)	dcl 148 set ref 153 180 180 181 184 188 194 198
icb	000113 automatic	fixed bin(17,0)	dcl 54 set ref 58 72 72 78 78 79 79 79 80 80 80 81 82 82 82 83 83 83 84 85 85 85 86 86 86 87 90 95 110 110 110 112 112 112 113 114 114 114 115 115 115 116 117 117 117 118 118 118 119
ifn	002104 automatic	fixed bin(17,0)	dcl 148 set ref 155 165 171 172 172 172 172 173
ilb	002103 automatic	fixed bin(17,0)	dcl 148 set ref 154 180 180 182 186 190 192 196
ilb	000114 automatic	fixed bin(17,0)	dcl 54 set ref 59 72 72 78 78 79 79 79 79 80 80 80 80 82 82 82 82 83 83 83 83 85 85 85 85 86 86 86 86 100 105 121 121 121 123 123 123 124 125 125 125 126 126 126 127 128 128 128 129 129 129 130
index	parameter	fixed bin(17,0)	dcl 6 set ref 1 8 25 30 44
iprop	000123 automatic	fixed bin(17,0)	dcl 6 set ref 21 24
iprop	parameter	fixed bin(17,0)	dcl 54 set ref 52 55 63 69 74
ires	parameter	fixed bin(17,0)	dcl 148 set ref 145 149 168
ires	000124 automatic	fixed bin(17,0)	dcl 6 set ref 28 29
j	002111 automatic	fixed bin(17,0)	dcl 148 set ref 159 160 160
k	000537 automatic	fixed bin(17,0)	dcl 227 set ref 292 294 295 295 296 297 307
k	002112 automatic	fixed bin(17,0)	dcl 148 set ref 157 160 160 162 162 163 163 163
l	parameter	float bin(27)	dcl 3 ref 1 13 19 35 35 36
lb	parameter	float bin(27)	dcl 53 ref 52 57 57 66 66 68
lb	parameter	float bin(27)	dcl 146 ref 145 151 151
lo	000106 automatic	float bin(27)	dcl 4 set ref 13 14 14 21 28
lbl	002101 automatic	float bin(27)	dcl 147 set ref 212 214 214
lbu	002100 automatic	float bin(27)	dcl 147 set ref 213 214
lcb	000107 automatic	float bin(27)	dcl 53 set ref 79 81 81 82 84 84 85 87 87 110 113 113 114 116 116 117 119 119
ll	000540 automatic	fixed bin(17,0)	dcl 227 set ref 307 308
llb	000111 automatic	float bin(27)	dcl 53 set ref 121 124 124 125 127 127 128 130 130
lll	002060 automatic	float bin(27)	dcl 146 set ref 200 208 208
llu	002061 automatic	float bin(27)	dcl 146 set ref 201 209 209
lul	002062 automatic	float bin(27)	dcl 146 set ref 202 208
luu	002063 automatic	float bin(27)	dcl 146 set ref 203 209
m	002113 automatic	fixed bin(17,0)	dcl 148 set ref 158 160 160 162 162 163 163 163
ncb	002106 automatic	fixed bin(17,0)	dcl 148 set ref 181 185 189 194 194 194 198 198 198 200 200 200 201 201 201 202 202 202 203 203 203 204 204 204 205 205 205 206 206 206 207 207 207 208 209 210 211
ncb	000115 automatic	fixed bin(17,0)	dcl 54 set ref 92 97 121 121 121 123 123 123 125 125 125 126 126 126 128 128 128 129 129 129
nfn	002107 automatic	fixed bin(17,0)	dcl 148 set ref 172 173 174 177 200 200 200 201 201 201 202 202 202 203 203 203 204 204 204 205 205 205 206 206 206 207 207 207
nlb	002105 automatic	fixed bin(17,0)	dcl 148 set ref 182 186 190 193 197 200 200 200 201 201 201 202 202 202 203 203 203 204 204 204 205 205 205 206 206 206 207 207 207 212 213
nlb	000116 automatic	fixed bin(17,0)	dcl 54 set ref 102 107 110 110 110 112 112 112 114 114 114 115 115 115 117 117 117 118 118 118
om	parameter	float bin(27)	dcl 219 ref 218 288 326
pdiam	000121 automatic	float bin(27)	dcl 5 set ref 33 49 49 50
propef	parameter	float bin(27)	dcl 3 set ref 1 50
propef	parameter	float bin(27)	dcl 219 set ref 218 325 327
q	000357 automatic	float bin(27)	dcl 219 set ref 320 324
qb1	000262 automatic	float bin(27)	dcl 219 set ref 269 281
qb2	000263 automatic	float bin(27)	dcl 219 set ref 270 281
qb3	000264 automatic	float bin(27)	dcl 219 set ref 271 281

qb4	000265 automatic	float bin(27)	dcl 219 set ref 272 284
qb5	000266 automatic	float bin(27)	dcl 219 set ref 273 284
qb6	000267 automatic	float bin(27)	dcl 219 set ref 274 285
qb7	000270 automatic	float bin(27)	dcl 219 set ref 275 286
qd	000361 automatic	float bin(27)	dcl 219 set ref 322 324
qf	000363 automatic	float bin(27)	dcl 219 set ref 324 325
qg1	000275 automatic	float bin(27)	dcl 219 set ref 281 351
qg2	000276 automatic	float bin(27)	dcl 219 set ref 282 351 353
qg3	000277 automatic	float bin(27)	dcl 219 set ref 283 351 353
qg4	000300 automatic	float bin(27)	dcl 219 set ref 284 351 353
qg5	000301 automatic	float bin(27)	dcl 219 set ref 285 352 354
qh1	000345 automatic	float bin(27)	dcl 219 set ref 322 351 356 357 359 362 386
qh2	000346 automatic	float bin(27)	dcl 219 set ref 322 352 357 359 362 365 386
qh3	000347 automatic	float bin(27)	dcl 219 set ref 286 322 359 362 365 367 386
qh4	000350 automatic	float bin(27)	dcl 219 set ref 353 356 357 359 362
qh5	000351 automatic	float bin(27)	dcl 219 set ref 354 357 359 362 365
r	000105 automatic	float bin(27)	dcl 4 set ref 47 48
rc	parameter	float bin(27)	array dcl 2 ref 1 160 162
rcb	000103 automatic	float bin(27)	dcl 53 set ref 56 56 58 81 84 87 91 96 113 116 119
rcb	002071 automatic	float bin(27)	dcl 147 set ref 150 150 153 208 209 210 211
rcbp	000105 automatic	float bin(27)	dcl 53 set ref 91 96 121 123 125 126 128 129
re	000372 automatic	float bin(27)	dcl 219 set ref 229 276
rfn	002073 automatic	float bin(27)	dcl 147 set ref 152 152 155 172 174 178
rk	000304 automatic	float bin(27)	dcl 219 set ref 288 355
rlb	000104 automatic	float bin(27)	dcl 53 set ref 57 57 59 79 80 82 83 85 86 101 106 124 127 130
rlb	002072 automatic	float bin(27)	dcl 147 set ref 151 151 154 212 213
rlbp	000106 automatic	float bin(27)	dcl 53 set ref 101 106 110 112 114 115 117 118
rn	000370 automatic	float bin(27)	dcl 219 set ref 330 356 356 357 357 357 357 359 359 359 359 359 359 362 362 362 362 362 362 362 362 365 365 365 365 365 365 367 367 367 367 369 369 386
rn	000116 automatic	float bin(27)	dcl 5 set ref 36 37 40
ro	000111 automatic	float bin(27)	dcl 5 set ref 17 47
rpm	parameter	float bin(27)	dcl 219 set ref 218 32[illegible]
rpm	000100 automatic	float bin(27)	dcl 3 set ref 50
sysprint	000016 constant	file	output stream dcl 7 set ref 7 11 43 61 62 67 68 166 167
t	parameter	float bin(27)	array dcl 2 set ref 1 82 82 82 83 83 83 114 114 114 115 115 115 125 125 125 126 126 126
t	000356 automatic	float bin(27)	dcl 219 set ref 319 323
tb1	000255 automatic	float bin(27)	dcl 219 set ref 264 277
tb2	000256 automatic	float bin(27)	dcl 219 set ref 265 277
tb3	000257 automatic	float bin(27)	dcl 219 set ref 266 278
tb4	000260 automatic	float bin(27)	dcl 219 set ref 267 279
tb5	000261 automatic	float bin(27)	dcl 219 set ref 268 279
td	000360 automatic	float bin(27)	dcl 219 set ref 321 323
temp	000104 automatic	float bin(27)	dcl 4 set ref 39 41
tf	000362 automatic	float bin(27)	dcl 219 set ref 323 325
tg1	000271 automatic	float bin(27)	dcl 219 set ref 277 347 349
tg2	000272 automatic	float bin(27)	dcl 219 set ref 278 347 349
tg3	000273 automatic	float bin(27)	dcl 219 set ref 279 348
tg4	000274 automatic	float bin(27)	dcl 219 set ref 280 348 350
th	parameter	float bin(27)	dcl 3 set ref 1 21 50
th	parameter	float bin(27)	dcl 219 ref 218 288 327
tn	parameter	float bin(27)	dcl 53 set ref 52 133
th1	000340 automatic	float bin(27)	dcl 219 set ref 287 321 356 357 359 362 372 375
th2	000341 automatic	float bin(27)	dcl 219 set ref 321 347 357 359 362 365 372 375 386

th3	000342 automatic	float bin(27)	dcl 219 set ref 321 348 359 362 365 367 372 375 386
th4	000343 automatic	float bin(27)	dcl 219 set ref 349 357 359 362 365
th5	000344 automatic	float bin(27)	dcl 219 set ref 350 359 362 365 367
tp	000101 automatic	float bin(27)	dcl 53 set ref 84 116 127 133
tq1	000305 automatic	float bin(27)	dcl 219 set ref 356 386
tq2	000306 automatic	float bin(27)	dcl 219 set ref 357 386
tq3	000307 automatic	float bin(27)	dcl 219 set ref 359 386
tq4	000310 automatic	float bin(27)	dcl 219 set ref 362 386
tq5	000311 automatic	float bin(27)	dcl 219 set ref 365 386
tq6	000312 automatic	float bin(27)	dcl 219 set ref 367 386
tq7	000313 automatic	float bin(27)	dcl 219 set ref 369 386
ucb	000110 automatic	float bin(27)	dcl 53 set ref 80 81 83 84 86 87 112 113 115 116 118 119
ulb	000112 automatic	float bin(27)	dcl 53 set ref 123 124 126 127 129 130
ull	002064 automatic	float bin(27)	dcl 146 set ref 204 210 210
ulu	002067 automatic	float bin(27)	dcl 146 set ref 205 211 211
uul	002065 automatic	float bin(27)	dcl 146 set ref 206 210
uuu	002066 automatic	float bin(27)	dcl 146 set ref 207 211
v	parameter	float bin(27)	dcl 3 set ref 1 15 50 167
vr	parameter	float bin(27)	dcl 219 ref 218 288 326
vs	000114 automatic	float bin(27)	dcl 5 set ref 15 19 36 47 48 49
w	parameter	float bin(27)	array dcl 2 set ref 1 79 79 79 80 80 80 110 110 110 112 112 112 121 121 121 123 123 123
wa	parameter	float bin(27)	dcl 219 ref 218 288 326 327
wa	parameter	float bin(27)	dcl 3 set ref 1 21 22 22 23 23 23 50
wa	parameter	float bin(27)	dcl 53 set ref 52 131
wp	000100 automatic	float bin(27)	dcl 53 set ref 81 113 124 131
wsa	000115 automatic	float bin(27)	dcl 5 set ref 35 47
x	parameter	float bin(27)	dcl 138 ref 137 139 139 140 140 141 141
x	parameter	float bin(27)	dcl 219 ref 218 230 230 230 231 231 232 232 232 235 235 236 239 240 240 241 242 242 244 245 245 246 246 247 247 247 247 248 248 248 249 249 249 249 249 251 251 251 252 252 253 254 255 255 256 257 258 259 259 259 260 262 264 265 266 267 268 268 269 270 271 274 275
x1	000100 automatic	float bin(27)	dcl 138 set ref 139 142
x2	000101 automatic	float bin(27)	dcl 138 set ref 140 142
x3	000102 automatic	float bin(27)	dcl 138 set ref 141 142
y	parameter	float bin(27)	dcl 138 set ref 137 142
y	000302 automatic	float bin(27)	dcl 219 set ref 291 301 304 311 331 331 331 331 332 332 332 332 333 334 334 335 335 335 336 336 336 337 338 338 339 339 339 339 340 340 340 341 341 341 342 342 342 342 343 343 343 344 344 345 345 346 346 346 347 347 348 349 350 351 351 351 352 353 353 371
y	002070 automatic	float bin(27)	dcl 146 set ref 174 178 200 201 202 203 204 205 206 207
y1	000330 automatic	float bin(27)	dcl 219 set ref 289 291 299 300 315
y2	000331 automatic	float bin(27)	dcl 219 set ref 290 299 300 309
ym	000332 automatic	float bin(27)	dcl 219 set ref 299 300 301 304 309 311 315
z	parameter	float bin(27)	dcl 329 set ref 328 385 386 386 386 386 386 386 386 386 386 386 386 386 386
z	000352 automatic	float bin(27)	dcl 219 set ref 293 319 319 319 320 320 320 321 321 322 322 325 326
z1	000353 automatic	float bin(27)	dcl 219 set ref 370 372 372 372 372 374 377 381
z2	000354 automatic	float bin(27)	dcl 219 set ref 371 374 377 379
zb	parameter	float bin(27)	dcl 219 ref 218 230 230 230 230 231 232 232 232 232 233 235 235 235 236 237 238 239 240 240 241 242 243 244 245 246 247 247 247 248 249 249 249

```
                                                                                      249 249 251 251 251 252 252 252 254 255 255 256
                                                                                      258 259 259 261 262 262 263 265 266 268 272 273
                                                                                      274 275 284
zb                              000120 automatic      float bin(27)                   dcl 5 set ref 34 49 50
zm                              000355 automatic      float bin(27)                   dcl 219 set ref 374 375 375 375 375 377 379 381
                                                                                      385

NAMES DECLARED BY EXPLICIT CONTEXT.
N801                            006643 constant       label                           dcl 374 ref 374 384
N805                            006731 constant       label                           dcl 385 ref 377 378 385
N809                            005177 constant       label                           dcl 299 ref 296 299 318
N810                            005221 constant       label                           dcl 306 ref 298 306
N811                            005164 constant       label                           dcl 293 ref 293 305
N840                            005223 constant       label                           dcl 307 ref 297 307
compl                           002444 constant       label                           dcl 131 ref 88 120 131
grend                           005374 constant       entry                           internal dcl 328 ref 293 328
next                            000601 constant       label                           dcl 45 ref 41 45
power                           000251 constant       entry                           external dcl 1 ref 1
prop                            004072 constant       entry                           internal dcl 218 ref 50 218
propfactors                     000703 constant       entry                           internal dcl 52 ref 21 52
quad                            002464 constant       entry                           internal dcl 137 ref 110 112 114 115 117 118 121
                                                                                      123 125 126 128 129 137 200 201 202 203 204 205
                                                                                      206 207
quad1                           002062 constant       label                           dcl 121 ref 93 98 121
quad2                           001470 constant       label                           dcl 110 ref 103 108 110
resist                          002525 constant       entry                           internal dcl 145 ref 28 145
zeld                            005250 constant       label                           dcl 319 ref 302 312 319

NAMES DECLARED BY CONTEXT OR IMPLICATION.
abs                                                   builtin function                internal ref 41 300 377
floor                                                 builtin function                internal ref 58 59 153 154 155
log10                                                 builtin function                internal ref 37 40 276
round                                                 builtin function                internal ref 10 14 20
sqrt                                                  builtin function                internal ref 19

STORAGE REQUIREMENTS FOR THIS PROGRAM.

          Object    Text      Link      Symbol    Defs      Static
Start          0       0      7200        7222    7101        7210
Length     14450    7101        22        5212      76           2

BLOCK NAME                    STACK SIZE     TYPE                WHY NONQUICK/WHO SHARES STACK FRAME
power                                509 external procedure      is an external procedure.
propfactors                          146 internal procedure      uses I/O statements.
quad                                  69 internal procedure      is called by several nonquick procedures.
resist                              1167 internal procedure      uses I/O statements.
prop                                     internal procedure      shares stack frame of external procedure power.
grend                                    internal procedure      shares stack frame of external procedure power.

THE FOLLOWING EXTERNAL OPERATORS ARE USED BY THIS PROGRAM.
fx1_to_fl2          r_l_a               r_g_a               r_e_as              r_ne_as             call_ext_out
call_int_this       call_int_other      return              fl2_to_fx1          ext_entry           int_entry
put_end             stream_io           floor_fl            round_fl            put_data_eis        put_list_eis
real_to_real_rd     sqrt                log10               real_p_real

THE FOLLOWING EXTERNAL ENTRIES ARE CALLED BY THIS PROGRAM.
decimal_exp_

THE FOLLOWING EXTERNAL VARIABLES ARE USED BY THIS PROGRAM.
sysprint                      sysprint.fsb
```

C.3 DICTIONARY OF VARIABLES USED IN THE BARGE POWERING MODEL

DEFINITION OF THE VARIABLES USED IN SUBROUTINE "power"

<u>Variable</u>	<u>Definition</u> (units)
b:	Barge Molded Breadth.
bt:	B/T. Ratio of barge molded breadth to barge draft.
c:	An array of values used as coefficients in Subroutine "prop".
cb:	Cb. Barge block coefficient.
cf:	Frictional Resistance Coefficient.
cr:	Residual Resistance Coefficient.
ct:	Total Resistance Coefficient.
d:	An array of values used as coefficients in Subroutine "prop".
dhp:	Delivered Horsepower. Horsepower required to be delivered to the propeller to provide the effective horsepower needed to propel barge at the specified speed.
dq:	An array of values used as coefficients in Subroutine "prop".
dr:	Barge Summer Waterline Draft.
dt:	An array of values used as coefficients in Subroutine "prop".
ehp:	Effective horsepower required to be delivered by the propeller to drive the barge at the specified speed.
eta:	Array of values of relative rotative efficiency coefficients used as input into Subroutine "propfactors".

exarea: Propeller Expanded Area Ratio. (sq in)

fn: Froude Number.

g: Acceleration Due to Gravity. (ft/sec)

gnu: Kinematic Velocity.

hr: Relative rotative efficiency for hull of specified form. Output of Subroutine "propfactors" and input into Subroutine "prop".

i: Index counter used in frictional resistance routine.

index: Subprogram "power" Error Index. Index that indicates whether calls to subroutines "propfactors" and "resist" have not been successful or that the frictional resistance routine does not converge (index=0 indicates success of Subroutine "power", index=1 indicates some error in the execution).

iprop: Subroutine "propfactors" Error Index. Index that indicates to Subroutine "power" that some error occurred during the execution of Subroutine "propfactors". (iprop=0 indicates no error; iprop=1 indicates error.)

ires: Subroutine "resist" Error Index. Index that indicates to Subroutine "power" that some error occurred during the execution of Subroutine "resist". (ires=0 indicates no error; ires=1 indicates error.)

l: Barge Length Between Perpindiculars.

lb: L/B. Ratio of barge length to barge molded breadth.

pdiam: Diameter of Propeller. (ft)

propef: Propeller Open Water Propulsive Efficiency. Output of Subroutine "prop".

rc: An array of values of residual resistance coefficients, used as input into Subroutine "resist".

r: Barge Resistance. Total still water resistance developed by barge.

rn:	Reynold's Number.
ro:	Density of Sea Water.
rpm:	RPM. Revolutions per minute of propeller at specified speed.
sysprint:	Output Switch Name.
t:	Array of values of thrust deduction fraction used as input into Subroutine "propfactors".
temp:	Temporary holding variable used in routine to calculate frictional resistance coefficient.
th:	Thrust deduction fraction for hull of specified form. Output of Subroutine "propfactors" and input into Subroutine "prop".
v:	Speed of Barge. (kts)
vs:	Speed of Barge. (ft/sec)
w:	Array of values of wake fraction used as input into Subroutine "propfactors".
wa:	Wake fraction for hull of specified form. Output of Subroutine "propfactors" and input into Subroutine "prop".
wsa:	Barge Wetted Surface Area.
zb:	Number of Blades on Propeller.

DEFINITION OF THE VARIABLES USED IN SUBROUTINE "propfactors"

Variable	Definition
cb:	Cb. Barge block coefficient.
etar:	Relative rotative efficiency for the specific barge form under consideration, multiplied by one thousand.
hr:	Relative rotative efficiency for the specific barge form under consideration.
icb:	Index used in the propulsion coefficient arrays indicating the first of two values with respect to cb that are used as inputs into the linear and quadratic interpolation routines for lb interpolation.
ilb:	Index used in the propulsion coefficient arrays indicating the first of two values with respect to lb that are used as inputs into the linear and quadratic interpolation routines for cb interpolation.
iprop:	Error Index. Index used to indicate whether lb or cb inputs are out of propulsion coefficient table ranges; (iprop=1 indicates inputs out of range; iprop=0 indicates inputs satisfactory).
lb:	L/B. Ratio of barge length to barge molded breadth.
lcb:	The linear or quadratic interpolated value with respect to lb for the lower of two cb values.
llb:	The quadratic interpolated value with respect to cb for the lower of two lb values.

ncb: Index used in propulsion coefficient arrays indicating the first of three cb table values that are inputted into the cb quadratic interpolation routine for a given lb value (ilb).

nlb: Index used in propulsion coefficient arrays indicating the first of three lb table values that are inputted into the lb quadratic interpolation routine for a given cb value (icb).

rcb: cb index value for which table interpolation is desired.

rcbp: Fractional part of cb index value for which table interpolation is desired and which is inputed into the cb quadratic interpolation routines.

rlb: lb index value for which table interpolation is desired.

rlbp: Fractional part of lb index value for which table interpolation is desired and which is inputted into the lb quadratic interpolation routines.

th: Thrust deduction fraction for the specific barge form under consideration.

tp: One thousand times one minus the thrust deduction fraction for the specific barge form under consideration.

ucb: The linear or quadratic interpolated value with respect to lb for the upper of two cb values.

ulb: The quadratic interpolated value with respect to cb for the upper of two lb values.

wa: The wake fraction for the specific barge form under consideration.

wp: One thousand times one minus the wake fraction for the specific barge form under consideration.

DEFINITION OF THE VARIABLES USED IN SUBROUTINE "resist"

Variable	Definition
bt:	B/T. Ratio of barge breadth to barge draft.
cb:	Cb. Barge block coefficient.
cbll:	Value of linear interpolation between two cb index values for the first of two lb index values, where bt = 2.46.
cblu:	Value of linear interpolation between two cb index values for the first of two lb index values, where bt = 2.76.
cbul:	Value of linear interpolation between two cb index values for the second of two lb index values, where bt = 2.46.
cbuu:	Value of linear interpolation between two cb index values for the second of two lb index values, where bt = 2.76.
cres:	Residual resistance coefficient found from the interpolation of the residual resistance table.
crp:	Array of residual resistance coefficients used for table interpolation. This array is identical to the input array rc except that 0.15 Froud number value is assumed to be linearly interpolated between the 0.14 and 0.16 Froude number values.
fn:	Froude Number.
i:	The bt index of the residual resistance array.
icb:	The integral part of the cb index value for which interpolation is desired.
ifn:	The integral part of the fn index value for which interpolation is desired.
ilb:	The integral part of the lb index value for which interpolation is desired.

ires: Index that indicates whether lb, cb, or fn inputs are out of table interpolation ranges (ires=1 indicates inputs are out of range; ires=0 indicates inputs satisfactory).

j: The Froude number (fn) index of the residual resistance coefficient array.

k: The cb index of the residual resistance coefficient array.

lb: L/B. Ratio of barge length to barge molded breadth.

lbl: Value of the linear interpolation between cbul and cbll, the two lb values, where bt = 2.46.

lbu: Value of the linear interpolation between cbuu and cblu, the two lb values, where bt = 2.76.

lll: The quadratically interpolated value with respect to fn for the first of two cb values and the first of two lb values, where bt = 2.46.

llu: The quadratically interpolated value with respect to fn for the first of two cb values and the second of two lb values, where bt = 2.46.

lul: The quadratically interpolated value with respect to fn for the second of two cb values and the first of two lb values, where bt = 2.46.

luu: The quadratically interpolated value with respect to fn for the second of two cb values and the second of two lb values, where bt = 2.46.

m: The lb index of the residual resistance coefficient array.

ncb: The cb index for the first of two table index values that bracket the index of the desired cb value.

nfn: The fn index for the first of three table index values that bracket the index of the desired fn value.

nlb: The lb index for the first of two table index values that bracket the index of the desired lb value.

rcb: The cb index for the desired cb value.

rfn: The fn index for the desired fn value.

rlb: The lb index for the desired lb value.

ull: The quadratically interpolated value with respect to fn for the first of two cb values and the first of two lb values, where bt = 2.76.

ulu: The quadratically interpolated value with respect to fn for the first of two cb values and the second of two lb values, where bt = 2.76.

uul: The quadratically interpolated value with respect to fn for the second of two cb values and the first of two lb values, where bt = 2.76.

uuu: The quadratically interpolated value with respect to fn for the second of two cb values and the second of two lb values, where bt = 2.76.

y: The fractional part of the index used to represent the desired fn value, used as input into the quadratic interpolation routine.

REFERENCES

American Bureau of Shipping (ABS)	1973	Rules for Building and Classing Steel Barges for Offshore Service, ABS, New York.
________	1978	Rules for Building and Classing Steel Vessels, ABS, New York.
Comstock, J. P. (ed.)	1967	Principles of Naval Architecture, SNAME, New York.
Evans, J. H. (ed.)	1975	Ship Structural Design Concepts, Cornell Maritime Press, Cambridge, Md.
Fletcher, E. H.	1974	"ARTUBAR--Resume of Achievements and Developments," author's report, August 1974.
Horne, M. R.	1958	"The Full Plastic Moments of Sections Subjected to Shear Force and Axial Load," British Welding Journal (April).
Hukill, E. L.	1974	"The Development of an Integrated Tug-Barge Concept," SNAME Technical Report, Toronto, March 1974.
IMCO	1966	"International Conference on Load Lines, 1966," Inter-Governmental Maritime Consultative Organization, London.
Jenkins, W. M. de Jesus, G. C. Burns, A.	1977	"Optimum Design of Welded Plate Girders," The Structural Engineer, Vol. 55, (April).
Maritime Administration (MARAD)	1979	From the program notebook of the National Ocean-Going Tug-Barge Planning Conference held on 27-28 March at New Orleans.

Robinson, J. H.	1976	"Experimental Research Relative to Improving the Hydrodynamic Performance of Oceangoing Tug/Barge Systems. Summary Report," NSRDC Ship Performance Department Report No. SPD-518-05 (NTIS No. PB 263 539).
Seabulk Chemical Carriers	1976	"Technical and Operational Features of Seabulk Magnachem/SCC 3902," company brochure.
Sharp, G. S.	1975	"Development of a Model for Economic and Operational Evaluation of Integrated Tug-Barge Systems," report for MARAD Contract No. 2-36285.
Triantafyllou, M. S.	1977	"Development of Analytical Methods in B-Series Propeller Design for Application in Computer Programs," Master Thesis M.I.T. Departments of Mechanical and Ocean Engineering.
Tscuchida, K. Koichi, Y. Yazaki, A. Shigeo, M. Seizo, O.	1969	"Systematic Resistance and Propulsion Tests With Models of Single-Screw Full-Bodied Oil Tankers," Japanese Ministry of Transport Ship Research Institute Report, Vol. 1, No. 6. Translated in Univ. of Michigan Depart. of N.A. and M.E. Report No. 039.
Waller, D. B. Filson, J. J.	1972	"Advanced Ocean Tug-Barge Systems: A Review of the State-of-the-Art of Ocean Barge Design and Operation," National Maritime Research Center Report No. NMRC-272-21000 RO (NTIS No. COM-74-10121).